

Desenvolupament d'una aplicació Android per la gestió d'un club de tennis i pàdel

Treball final de grau

Grau: Enginyeria de sistemes TIC

Autor: David Bruch Batlló

Tutora: Marta Isabel Tarrés Puertas

Curs: 2017 / 2018

Agraïments

En primer lloc donar les gràcies a la meva família, concretament al meu pare que ha ajudat a fer possible que pugui acabar fent aquest grau. També a la meva tutora, la Marta Isabel Tarrés Puertas, la qual m'ha ajudat en dubtes sobre la base de dades i com enfocar el projecte en general. Donar les gràcies als meus companys i especialment al Josep Nuño que ha sigut un bon company de pràctiques i un bon amic.

Finalment dir que durant tot el transcurs d'aquest projecte he tingut problemes oculars que actualment s'estan tractant però que m'han dificultat una mica el procés. Donar gràcies per la comprensió dels companys i professors.

Resum

El projecte consisteix en el desenvolupament d'una aplicació mòbil per *Android* ajustada al control d'un club de tennis i pàdel com el de Navàs.

L'aplicació disposa de funcionalitats separades segons si ets un usuari normal o un administrador. Les funcionalitats dels usuaris normals permeten registrar-se al club, reservar hores, participar en torneigs, consultar el temps atmosfèric, etc. Per altra banda, els administradors poden controlar totes les interaccions dels usuaris pel que fa a reserves i torneigs, còmodament des de l'aplicació per tenir un control absolut del club.

Totes les funcionalitats estan pensades per millorar la gestió del club i agilitzar-la fent a la gent més fàcil jugar a tennis i a pàdel.

Abstract

This project consists in the development of a mobile application for *Android* adjusted to control a tennis and paddle club like Club Tennis Navàs.

The application has separate features depending on whether you are a normal user or an administrator. The user's normal functionality allows you to registre into the club, reserve hours, participate in tournaments, check the weather, etc. On the other hand, the administrators can control all user interactions when making reservations and tournaments comfortably from the application to have absolute control of the club.

All the features are designed to improve club management and make it easier for people to play tennis and paddle.

Paraules claus

Al llarg de la memòria del projecte s'utilitzen un conjunt de paraules tècniques, les quals, s'expliquen a continuació.

Smartphone: Anglisme. Telèfon mòbil construït sobre una plataforma informàtica mòbil, amb capacitat per emmagatzemar dades i realitzar activitats assemblant-se a un miniordinador.

Android: Sistema operatiu basat en linux per a dispositius mòbils.

Activitat: Cadascuna de les pantalles de l'aplicació mòbil.

Fragment: Representa una secció o part d'una activitat que és intercanviable.

Layout: defineix l'estructura visual per a una interfície d'usuari d'una pantalla de l'aplicació.

Preferència: Mecanisme d'emmagatzematge més senzill d'*Android* en forma de parts clau-valor.

API: Sigles d'Interfície de Programació d'Aplicacions. És un conjunt de funcions, mètodes o procediments que ofereix una biblioteca per ser utilitzat per un altre programari com una capa d'abstracció.

Servei Web: Tecnologia que utilitza un conjunt de protocols i estàndards que serveixen per intercanviar dades entre aplicacions.

Recurs: Dades (arrays, imatges, cadenes, números, estils) que s'emmagatzemen independentment del codi d'aplicació per millorar l'organització del projecte i permetre la seva adaptació a canvis.

Client: Aplicació informàtica que consumeix un servei remot d'un altre servei conegut com a servidor.

Servidor: Node que forma part d'una xarxa i proveeix de serveis a nodes anomenats clients.

Script: Arxiu d'ordres, arxiu de processament per lots o guió és un programa usualment simple, que regularment s'emmagatzema en arxius de text pla.

Multiplataforma: Vàlid per a diversos sistemes.

SDK: paquet d'eines pel desenvolupament d'aplicacions d'*Android*.

Shared preferences: permet emmagatzemar informació en forma de clau-valor en les aplicacions.

Singleton: permet restringir la creació d'objectes que pertanyen a un valor o un únic tipus d'objecte per tal de garantir una única instància de la classe i proporcionar accés global a ella.

Índex

1.	Introducció	1
1.1.	Context i justificació del treball	1
1.2.	Objectius.....	1
1.3.	Objectius del projecte	1
1.4.	Fases de realització.....	2
2.	Estat de l'art.....	3
2.1.	<i>Android</i>	3
2.2.	Enfocaments existents	5
2.2.1.	“Coge pista”	5
2.2.2.	“Club de Tennis Chamartín”	5
2.2.3.	“Playtomic - Reserva pista de pádel y tenis”	6
3.	Anàlisis	6
3.1.	Catàleg de requisits.....	6
3.1.1.	Requisits funcionals.....	6
3.1.2.	Requisits no funcionals.....	10
3.2.	Metodologia	11
3.3.	Seguretat del projecte	11
3.3.1.	Seguretat de la informació guardada	12
3.3.2.	Seguretat en la comunicació de la informació	13
3.3.3.	Seguretat per accedir a la base de dades remota	13
4.	Disseny	15
4.1.	Arquitectura de l'aplicació	15
4.2.	Tecnologies utilitzades i eines de treball.....	15
4.2.1.	<i>Android Studio</i>	16
4.2.2.	XAMPP	17
4.2.3.	Servidor HTTP Apache.....	17
4.2.4.	SQLyog.....	17
4.2.5.	MySQL.....	18
4.2.6.	000webhost	19
4.3.	Llenguatges de programació	20
4.3.1.	Java	20
4.3.2.	PHP	20
4.3.3.	<i>Python</i>	20
4.4.	Libreries externes utilitzades	21
4.4.1.	<i>Volley</i>	21
4.4.2.	<i>PHPMailer</i>	21
4.4.3.	Password Hash	22
4.4.4.	MySQLi	22
4.4.5.	PyOWM	22

4.4.6.	Requests	23
4.4.7.	Time.....	23
4.5.	Disseny del software de l'aplicació <i>Android</i>	23
4.5.1.	Disseny de la interfície gràfica	23
4.5.2.	Classes Java de l'aplicació <i>Android</i>	24
4.6.	Disseny de la base de dades	27
4.6.1.	Disseny de les taules a la base de dades	27
4.6.2.	Model entitat-relació	28
4.6.3.	Creació i format de les taules	30
4.7.	Disseny de les peticions HTTP i gestió al servidor	36
4.8.	Disseny de “scripts” i procediments importants	38
4.8.1.	Disseny de les reserves	38
4.8.2.	Disseny dels torneigs	38
4.8.3.	Disseny de modificació de partits	42
4.8.4.	Disseny del programa de “pyhton” per registrar el temps atmosfèric	43
4.9.	Disseny de la seguretat	44
4.9.1.	Disseny de la seguretat del client.....	44
4.9.2.	Disseny de la seguretat del servidor i la base de dades	48
5.	Implementació	50
5.1.	Part del client.....	50
5.1.1.	Menú lateral (<i>Navegation Drawer</i>)	50
5.1.2.	Arxiu Manifest.....	52
5.1.3.	Arxiu build.gradle.....	52
5.1.4.	Directorí assets.....	53
5.1.5.	Directorí res	53
5.1.6.	Permisos de la càmera, lectura i escriptura	54
5.1.7.	Interaccions amb els elements de la interfície gràfica	54
5.1.8.	Comunicació amb el servidor utilitzant <i>Volley</i>	55
5.1.9.	<i>Shared preferences</i>	58
5.1.10.	Tasques asíncrones	58
5.2.	Part del servidor	59
5.2.1.	Gestió de la base de dades remota	59
5.2.2.	Fitxers del servidor	59
5.2.3.	Enviament de correus amb <i>PHPMailer</i>	60
5.3.	Part externa	61
5.3.1.	Programa pyhton de registre atmosfèric	61
6.	Proves	62
6.1.	Casos de prova	62
6.1.1.	Casos de prova per usuaris normals.....	62
6.1.2.	Casos de prova per administradors	66

7.	Pressupost de l'aplicació	69
8.	Conclusions	70
9.	Treball futur	70
10.	Referències utilitzades	71
10.1.	Il·lustracions extretes d'internet.....	71
10.2.	Referències web	72
Annex A. Manual d'ús de l'aplicació.....		73
Annex B: Funcionalitats d'Android		112

Índex d'il·lustracions

Il·lustració 1: Diagrama del ús de les versions d'Android. Font: Externa.....	4
Il·lustració 2: "Coge pista". Font: Externa.....	5
Il·lustració 3: "Club de Tenis Chamartín". Font: Externa.....	5
Il·lustració 4: "Playtomic - Reserva pista de pádel y tennis". Font: Externa	6
Il·lustració 5: Parts diferenciades del algoritme de xifratge Bcrypt. Font: Externa	13
Il·lustració 6: Arquitectura del sistema	15
Il·lustració 7: Logotip d'Android Studio. Font: Externa	16
Il·lustració 8: Logotip de XAMPP. Font: Externa.....	17
Il·lustració 9: Logotip d'Apache. Font: Externa	17
Il·lustració 10: Logotip de SQLyog. Font: Externa	18
Il·lustració 11: Logotip de MySQL. Externa.....	18
Il·lustració 12: Logotip de 000webhost. Externa	19
Il·lustració 13: Informació del servidor web.....	19
Il·lustració 14: Informació del gestor phpMyAdmin	19
Il·lustració 15: Informació del servidor de base de dades.....	20
Il·lustració 16: Logotip de PHPMailer. Font: Externa	21
Il·lustració 17: Logotip de la llibreria pyown. Font: Externa	22
Il·lustració 18: Logotip de OpenWeatherMap. Font: Externa.....	22
Il·lustració 19: Pantalla d'inici de sessió.....	23
Il·lustració 20: Vista global de totes les taules de la base de dades.....	28
Il·lustració 21: Esquema entitat-relació de la gestió de reserves.....	28
Il·lustració 22: Esquema entitat-relació de la gestió de torneigs.....	29
Il·lustració 23: Taules per altres gestions.....	30
Il·lustració 24: Informació de la taula d'usuaris	30
Il·lustració 25: Informació de la taula d'administradors	31
Il·lustració 26: Informació de la taula de reserves.....	31
Il·lustració 27: Informació de la taula d'equips	31
Il·lustració 28: Informació de la taula pertinença.....	32
Il·lustració 29: Informació de la taula de partits.....	32
Il·lustració 30: Informació de la taula de torneigs.....	32
Il·lustració 31: Informació de la taula de registres.....	33
Il·lustració 32: Informació de l'historial de torneigs	33
Il·lustració 33: Informació de la taula de la llista negra	33
Il·lustració 34: Informació de la taula del registre atmosfèric.....	34
Il·lustració 35: Esquema de la base de dades generat per phpMyAdmin	35
Il·lustració 36: Codi del fitxer Consulta.php.....	36
Il·lustració 37: Codi del fitxer Modifica.php	37
Il·lustració 38: Evolució dels partits en un torneig de 8.....	41
Il·lustració 39: Diagrama de comunicacions del programa de "pyhton" i el servidor	43
Il·lustració 40: Imatge de el correu electrònic d'activació del compte.....	44
Il·lustració 41: Missatge advertint que no s'ha produït cap canvi	45
Il·lustració 42: Missatge confirmant que s'ha enviat el correu	45
Il·lustració 43: Correu electrònic de confirmació del canvis	46
Il·lustració 44: Xifratge de la contrasenya	46
Il·lustració 45: Verificació de la contrasenya	46
Il·lustració 46: Missatge avisant que el "nick" esta buit.....	47
Il·lustració 47: Diàleg de confirmació per la contrasenya de recuperació	47
Il·lustració 48: Missatge d'avís de que no existeix un usuari amb aquest "nick"	47
Il·lustració 49: Missatge de que s'ha enviat el correu correctament	48
Il·lustració 50: Protecció dels fitxer PHP amb contrasenya activada.....	49
Il·lustració 51: Part de codi necessària per autenticar les peticions	49
Il·lustració 52: Exemple de que demana autenticació des del navegador	49
Il·lustració 53: Validació del certificat SSL per la petició HTTPS	50
Il·lustració 54: Exemple per veure que la connexió és HTTPS.....	50

Il·lustració 55: Menú desplegable lateral.....	51
Il·lustració 56: Implementació del menú desplegable a l'activitat principal.....	51
Il·lustració 57: Generació del menú lateral desplegable	51
Il·lustració 58: Funció que captura l'opció seleccionada.....	51
Il·lustració 59: Parts de la funció que tria el fragment depenen de la id	52
Il·lustració 60: Permisos del arxiu Manifest.....	52
Il·lustració 61: Codi principal del arxiu build.gradle.....	53
Il·lustració 62: Dependències utilitzades.....	53
Il·lustració 63: Funció que gestiona els permisos de la càmera, de lectura i escriptura.....	54
Il·lustració 64: Exemple de l'associació d'un botó.....	54
Il·lustració 65: Funció per comprovar que el DNI és correcte	55
Il·lustració 66: Funció per comprovar que el correu electrònic és correcte	55
Il·lustració 67: Petició GET 1.....	55
Il·lustració 68: Petició GET 2.....	56
Il·lustració 69: Petició GET 3.....	56
Il·lustració 70: Petició GET 4.....	56
Il·lustració 71: Petició POST 1	56
Il·lustració 72: Petició POST 2	57
Il·lustració 73: Petició POST 3	57
Il·lustració 74: Petició POST 4	57
Il·lustració 75: Petició POST 5	58
Il·lustració 76: Emmagatzematge del Shared Preferences.....	58
Il·lustració 77: Implementació de AsyncTask.....	58
Il·lustració 78: Gestió de la base de dades des de 000webhost.....	59
Il·lustració 79: Menú de control de phpMyAdmin	59
Il·lustració 80: Connexió amb la base de dades	59
Il·lustració 81: Sentència SQL.....	60
Il·lustració 82: Enviament de la resposta en format JSON	60
Il·lustració 83: Exemple de modificació.....	60
Il·lustració 84: Codi per enviar correus electrònics des del servidor.....	61
Il·lustració 85: Funció del temps actual	61
Il·lustració 86: Funció de la petició post.....	62
Il·lustració 87: Funció de la crida d'hores.....	62
Il·lustració 88: Menú principal i menú desplegable lateral	73
Il·lustració 89: Missatge d'avís per iniciar sessió o registrar	74
Il·lustració 90: Menú desplegable sens usuari	74
Il·lustració 91: Pantalla d'inici de sessió buida i omplerta	74
Il·lustració 92: Quadre de diàleg per confirmar el tancament de sessió	75
Il·lustració 93: Quadre de diàleg per confirmar l'eliminació del compte.....	75
Il·lustració 94: Pantalla de registre buida	76
Il·lustració 95: Quadre d'opcions per importar la imatge d'usuari.....	76
Il·lustració 96: Pantalla de registre omplerta.....	77
Il·lustració 97: Missatge de confirmació del registre	77
Il·lustració 98: Pantalla per editar la informació d'un usuari	78
Il·lustració 99: Quadre d'opcions per importar la imatge d'usuari.....	78
Il·lustració 100: Missatge del fet que s'ha enviat un correu de confirmació	79
Il·lustració 101: Correu electrònic de confirmació	79
Il·lustració 102: Calendari disponible per triar el dia de la reserva	80
Il·lustració 103: Pantalla de reserva.....	80
Il·lustració 104: Pantalles de reserva amb hores ocupades per un altre usuari, per l'usuari actual i hores clicades per reservar-ne de noves.	81
Il·lustració 105: Procés d'eliminació de les reserves d'una pista	82
Il·lustració 106: Correu electrònic del recordatori de reserva.	82
Il·lustració 107: Pantalla de reserves actives	83
Il·lustració 108: Missatge de cap reserva activa que porta al menú principal	83
Il·lustració 109: Botó per eliminar i diàleg de confirmació.....	83

Il·lustració 110: Pantalla d'inici a l'entrar al torneig de tennis	84
Il·lustració 111: Pantalla per la creació del torneig de tennis.....	85
Il·lustració 112: Pantalla del torneig de tennis	86
Il·lustració 113: Requadre per introduir el nom d'equip	86
Il·lustració 114: Equip registrat correctament.....	87
Il·lustració 115: Llista d'equips al torneig de tennis.....	87
Il·lustració 116: Arbre de partits del torneig de tennis en una etapa inicial.....	88
Il·lustració 117: Arbre de partits del torneig de tennis en una etapa mitjana i final	89
Il·lustració 118: Taula de partits del club de tennis	89
Il·lustració 119: Pantalla d'inici a l'entrar al torneig de pàdel	90
Il·lustració 120: Pantalla per la creació d'un torneig de pàdel	91
Il·lustració 121: Pantalla del torneig de pàdel	92
Il·lustració 122: Requadre per introduir el segon integrant	92
Il·lustració 123: Requadre per introduir el nom d'equip de pàdel	93
Il·lustració 124: Llista d'equips al torneig de pàdel	93
Il·lustració 125: Arbre de partits del torneig de pàdel en una etapa inicial	94
Il·lustració 126: Taula de partits del torneig de pàdel	95
Il·lustració 127: Pantalla de l'historial de torneigs.....	96
Il·lustració 128: Pantalla del registre atmosfèric	96
Il·lustració 129: Pantalla d'ubicació del club	97
Il·lustració 130: Pantalla del temps atmosfèric al club	97
Il·lustració 131: Pantalla per contactar amb el club	98
Il·lustració 132: Pantalla del menú d'administradors.....	98
Il·lustració 133: Llista d'usuaris	99
Il·lustració 134: Llista d'administradors	100
Il·lustració 135: Pantalla inicial per expulsar un usuari del club.....	100
Il·lustració 136: Pantalla d'expulsió amb l'spinner obert i la raó omplerta	101
Il·lustració 137: Llista de totes les reserves actives	101
Il·lustració 138: Calendari i pantalla per ocupar hores sense limitacions	102
Il·lustració 139: Correu avisant que la reserva ha sigut cancel·lada	102
Il·lustració 140: Quadre de text per confirmar ocupar un dia sencer	103
Il·lustració 141: Quadre de text per confirmar la desocupació del dia sencer	103
Il·lustració 142: Exemple del correu d'avís de cancel·lació de reserves	104
Il·lustració 143: Pantalla de la llista negra.....	104
Il·lustració 144: Pantalla que permet treure un usuari de la llista negra	105
Il·lustració 145: Pantalla per enviar missatges de difusió	106
Il·lustració 146: Diàleg per confirmat l'enviament als administradors	106
Il·lustració 147: Diàleg per confirmar l'enviament als usuaris normals	106
Il·lustració 148: Pantalla per ocupar o buidar un torneig	107
Il·lustració 149: Pantalla per omplir el torneig	108
Il·lustració 150: Pantalla per buidar el torneig seleccionat anteriorment	109
Il·lustració 151: Pantalla per modificar els partits de qualsevol torneig actiu	110
Il·lustració 152: Primera zona per modificar partits.....	110
Il·lustració 153: Segona zona no habilitada i habilitada.....	111
Il·lustració 154: Tercera zona no habilitada i habilitada.....	111
Il·lustració 155: Diagrama d'estats importants d'una activitat. Font: Externa	113
Il·lustració 156: Cicle de vida d'un fragment. Font: Externa	114
Il·lustració 157: Esquema d'una activitat que conté 2 fragments. Font: Externa	115
Il·lustració 158: Exemple d'un botó i un boto imatge	115
Il·lustració 159: Exemple de un text pla i un quadre de text	115
Il·lustració 160: Exemple de RadioButton i d'Spinner	116
Il·lustració 161: Exemple de 2 diàlegs diferents.....	116

Índex de taules

Taula 1: Historial de versions d'Android. Font: Externa.....	3
Taula 2: Exemple del xifratge amb els 3 possibles algoritmes	13
Taula 3: Classes Java sobre la interfície gràfica.....	25
Taula 4: Funcions de la classe equip individual.....	25
Taula 5: Funcions de la classe equip parella	25
Taula 6: Funcions de la classe partit.....	25
Taula 7: Funcions de la classe torneig.....	26
Taula 8: Funcions de la classe usuari	26
Taula 9: Fitxers PHP per propòsits específics	38
Taula 10: Cas de prova del registre	63
Taula 11: Cas de prova del inici de sessió.....	63
Taula 12: Cas de prova per editar perfil.....	64
Taula 13: Cas de prova per recuperar contrasenya.....	64
Taula 14: Cas de prova per donar de baixa un compte.....	64
Taula 15: Cas de prova per reservar una pista.....	65
Taula 16: Cas de prova per eliminar una reserva	65
Taula 17: Cas de prova del registre d'un torneig	66
Taula 18: Cas de prova per modificar un partit.....	66
Taula 19: Cas de prova per eliminar un usuari	67
Taula 20: Cas de prova per ocupar hores de reserva	67
Taula 21: Cas de prova per ocupar dies de reserva	67
Taula 22: Cas de prova per treure un usuari de la llista negra	67
Taula 23: Cas de prova del missatge de difusió	68
Taula 24: Cas de prova per buidar un torneig.....	68
Taula 25: Cost de la mà d'obra	69
Taula 26: Cost del software.....	69
Taula 27: Cost del hardware	69
Taula 28: Cost total del projecte.....	70

1. Introducció

1.1. Context i justificació del treball

Al llarg dels últims anys, l'avanç de la tecnologia per dispositius mòbils ha sigut tal que cada cop més hi ha un gran nombre d'accions que han sigut absorbides completament per l'ús dels telèfons mòbils intel·ligents i internet. Des de la compra de productes per internet fins a la comunicació instantània entre dues persones. Cada dia surten a la llum noves aplicacions cada cop més útils, fàcils i còmodes d'utilitzar que ens fan possibles la realització de tasques quotidianes des dels palmells de les nostres mans.

És indubtable que els avantatges que això suposa superen els inconvenients. Aquest tipus de tecnologia ha ajudat a millorar molts aspectes de la nostra vida quotidiana, com per exemple amb l'accés a la informació, l'emmagatzematge de dades personals de forma segura o la compra de manera immediata, senzilla i segura des de qualsevol lloc.

El desenvolupament del treball de final de grau és una gran oportunitat per realitzar una aplicació en *Android*. Per la realització del projecte s'ha buscat una aplicació que sigui útil i interessant. Finalment, després de pensar en diferents opcions m'he decidit per la gestió d'un club de tennis i pàdel que són dos esports que van molt lligats de la mà, que m'agraden molt i que cada cop més gent gaudeix.

Sempre és important pels jugadors saber amb exactitud les hores lliures que tenen per reservar una pista per jugar i poder reservar de manera ràpida i senzilla. A més a més, també s'han de poder cancel·lar hores de la mateixa manera i poder consultar els partits i l'avanç d'un torneig. Per aquesta raó el desenvolupament d'aquesta aplicació vol ajudar a poder gestionar tots aquests factors per aconseguir una millor comoditat i eficiència en la gestió d'un club.

1.2. Objectius

Com ja he comentat una mica a l'apartat anterior l'objectiu principal és el desenvolupament d'una aplicació mòbil per la gestió d'un club de tennis i pàdel, que ve motivada per la necessitat que tenen molts clubs de poder tenir una gestió a temps real de les reserves de pistes i dels torneigs sense necessitat que hi hagi d'haver algú al darrere d'un telèfon controlant trucades i apuntant les reserves. D'aquesta manera, es pot tenir una gestió molt més eficaç del club tant pels treballadors del club com pels usuaris que vulguin utilitzar els seus serveis.

1.3. Objectius del projecte

Els objectius principals d'aquesta aplicació seran els següents:

1) Gestió i control dels usuaris registrats a l'aplicació.

- Registre segur d'usuaris mitjançant correu electrònic de verificació.
- Que sigui obligatori guardar una imatge d'usuari per poder identificar a les persones en els torneigs.
- Inici de sessió en un compte creat prèviament, necessari per poder utilitzar les altres funcionalitats de l'aplicació.
- Xifratge de la contrasenya de tots els usuaris per més seguretat.
- Recuperació de la contrasenya utilitzant un correu i una contrasenya de recuperació.
- Poder editar qualsevol informació d'un usuari amb la seguretat d'haver d'acceptar els canvis des del correu.

2) Gestió i control de les reserves de pistes.

- Reservar una pista concreta, en un dia concret i unes hores concretes tenint en compte l'usuari que ha iniciat sessió per tenir control i registre de totes les reserves.

- Veure les hores ocupades per un mateix usuari i les hores ocupades pels altres usuaris.
 - Poder reservar les hores lliures i poder cancel·lar la reserva en qualsevol moment.
 - Cancel·lar totes les reserves d'un mateix usuari o només les reserves d'un dia concret.
- 3) Gestió i control de torneigs temporals de tennis i pàdel.**
- Crear un torneig amb diferents característiques. Funcionalitat que només podran dur a terme els administradors.
 - Registrar-se a un torneig com un equip que pot ser individual (tennis) o per parelles (tennis o pàdel).
 - Omplir el torneig fins al màxim de jugadors disponibles en el torneig actual.
 - Iniciar el torneig i que es generin automàticament els partits entre equips de la primera ronda.
 - Fer avançar el torneig a través de les interaccions dels diferents usuaris dels equips posant els resultats dels partits i qui ha guanyat o perdut.
 - Donar un guanyador final i poder finalitzar el torneig temporal per poder crear-ne un altre quan es vulgui.
 - Poder veure l'històric de torneigs acabats amb la informació de la durada i les característiques d'aquests.
- 4) Localització i temps atmosfèric.**
- Poder veure la localització del club a través de *Google maps* o alguna pàgina semblant.
 - Poder veure el temps que farà en el club a través de *Meteocat* o pàgines semblants.
 - Poder veure el temps a la pista en temps real amb una aplicació externa.
- 5) Control dels administradors**
- Poder veure una llista de tots els usuaris i la seva informació.
 - Poder eliminar un usuari posant-lo a una "llista negra" perquè no pugui registrar-se novament a no ser que un administrador li permeti.
 - Funcionalitat per veure totes les reserves actives amb la informació de qui ha reservat i de quines hores.
 - Funcionalitat de poder ocupar tant hores concretes com dies, que tindrà prioritat davant d'una reserva normal i a més a més avisarà pel correu als usuaris que se'ls hi hagi ocupat l'hora.
 - Poder veure la llista d'expulsats i la seva informació per si es volen admetre novament en un futur o no.
 - Funcionalitat per treure algú de la "llista negra" perquè pugui tornar a registrar-se.
 - Funcionalitat per poder fer un missatge de difusió a tots els usuaris o a tots els administradors (per si són qüestions internes del club).
 - Funcionalitat per poder omplir o buidar un torneig creat que encara no s'ha iniciat perquè s'està omplint.
 - Funcionalitat per poder modificar els partits per separat de qualsevol torneig actiu.

1.4. Fases de realització

En aquest apartat es vol donar una visió general de les fases seguides al llarg de la realització del projecte, que són les corresponents a la majoria de desenvolupaments de software.

- 1) **Estudi:** Investigació sobre les possibles tecnologies a utilitzar. Indagació i proves sobre aplicacions semblants amb l'objectiu d'extreure el que se sol utilitzar en projectes similars.
- 2) **Anàlisi:** Definició dels requisits amb l'objectiu de veure exactament que és el que es vol construir.

- 3) **Disseny:** Després d'acabar la fase d'estudi i anàlisi i amb el llistat de funcions i requisits ben definits, es passa a dissenyar l'arquitectura de l'aplicació. Això inclou diferents diagrames que definiran el disseny de pantalles, interfície, bases de dades i seguretat.
- 4) **Implementació:** Programació de l'aplicació mòbil en *Android Studio* (JAVA) i del servidor amb PHP.
- 5) **Proves:** Un cop finalitzades totes les fases anteriors, es fa necessari la realització d'una fase de proves generals per comprovar la qualitat de l'efectivitat del software desenvolupat.
- 6) **Documentació:** La documentació és imprescindible pel correcte manteniment de qualsevol producte de software. Tant en l'àmbit de codi com de document amb el fi de deixar constància per escrit de les fases dutes a terme en el cicle de vida d'aquest producte software.
- 7) **Manteniment:** El manteniment d'un producte software suposa el 80% del seu cicle de vida per al que és de gran importància perquè no es quedi obsolet abans d'hora.

2. Estat de l'art

2.1. Android

Android és un sistema operatiu basat en el *kernel* de *Linux* i dissenyat principalment per dispositius mòbils amb pantalla tàctil, tals com *smartphones* o tablettes. Inicialment desenvolupat per *Android Inc*, i ajudat econòmicament per *Google*, que més tard, l'any 2005 va adquirir l'empresa.

Un dels aspectes fonamentals del sistema operatiu d'*Android* va ser la seva orientació a la *multiplataforma*, una característica realment innovadora, perquè el normal abans era que un sistema operatiu s'associés correctament només a un únic dispositiu. Ràpidament aquesta característica va fer que *Android* aconseguís ser el sistema operatiu més utilitzat per telèfons mòbils.

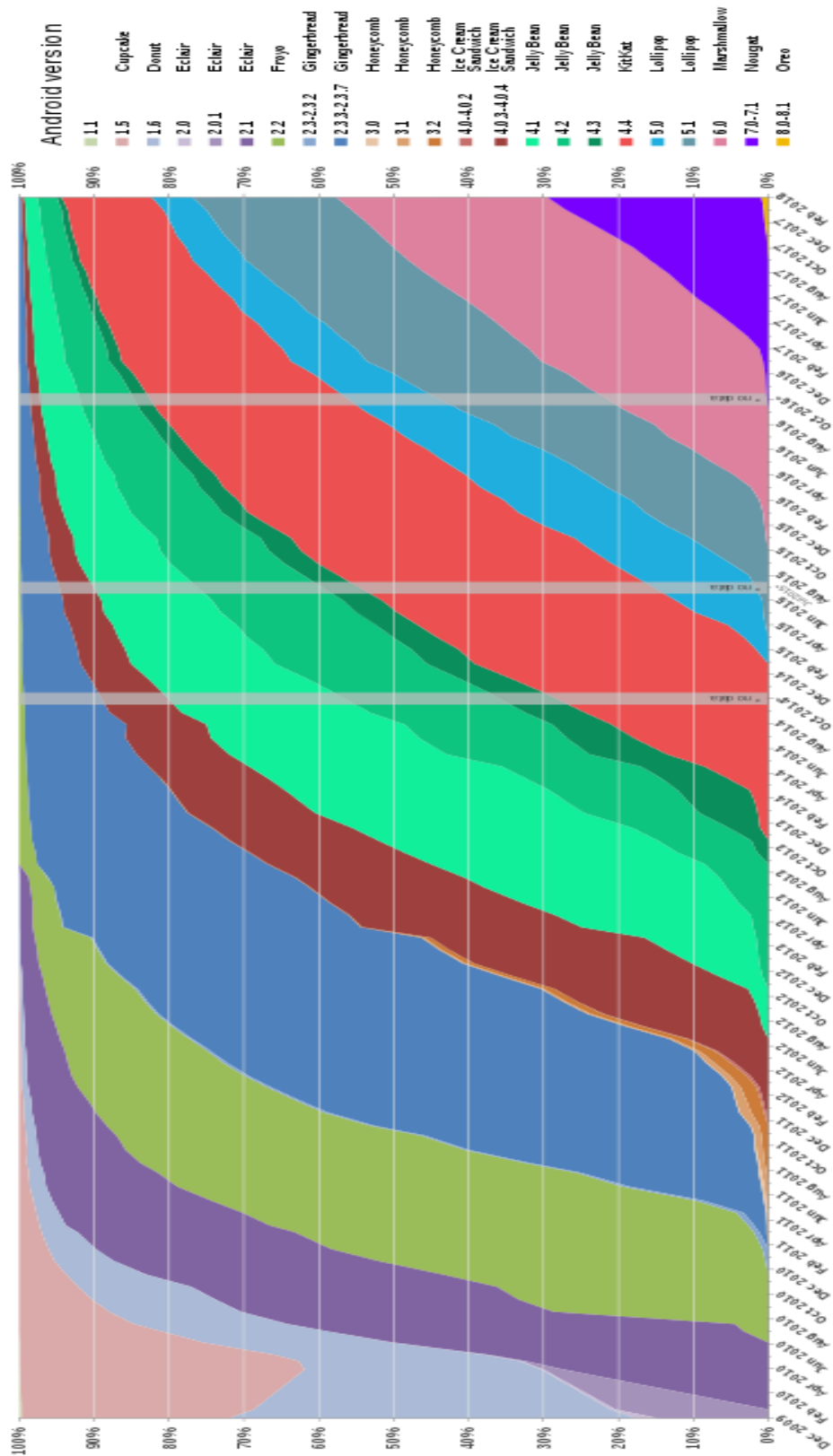
El sistema d'*Android* disposa de diferents versions des de la seva creació on els seus noms segueixen l'ordre de l'abecedari arribant a la lletra P.

Nom en clau	Versió	Data de llançament	Nivell de l'API	Suport
	1.0	23 de Setembre de 2008	1	No
	1.1	9 de Febrer de 2009	2	No
Cupcake	1.5	27 d'Abril de 2009	3	No
Donut	1.6	15 de Setembre de 2009	4	No
Eclair	2.0 - 2.1	26 d'Octubre de 2009	5 - 7	No
Froyo	2.2 - 2.2.3	20 de Maig de 2010	8	No
Gingerbread	2.3 - 2.3.7	6 de Desembre de 2010	9 - 10	No
Honeycomb	3.0 - 3.2.6	22 de Febrer de 2011	11 - 13	No
Ice Cream Sandwich	4.0 - 4.0.4	18 d'Octubre de 2011	14 - 15	No
Jelly Bean	4.1 - 4.3.1	9 de Juliol de 2012	16 - 18	No
KitKat	4.4 - 4.4.4	31 d'Octubre de 2013	19 - 20	No
Lollipop	5.0 - 5.1.1	12 de Novembre de 2014	21 - 22	No
Marshmallow	6.0 - 6.0.1	5 d'Octubre de 2015	23	Sí
Nougat	7.0 - 7.1.2	22 d'Agost de 2016	24 - 25	Sí
Oreo	8.0 - 8.1	21 d'agost de 2017	26 - 27	Sí
Pie	9.0	6 d'agost de 2018	28	Sí

Llegenda: Versió antiga / Darrera versió / Darrera versió preliminar

Taula 1: Historial de versions d'Android. Font: [Externa](#)

En el següent diagrama es pot observar l'evolució de l'ús de les versions *Android* des del desembre del 2009 fins al febrer del 2018 on es pot veure clarament com actualment les versions més utilitzades són les últimes que han sortit.



Il·lustració 1: Diagrama del ús de les versions d'Android. Font: [Externa](#)

2.2. Enfocaments existents

Pel desenvolupament d'una aplicació d'*Android*, existeixen 2 plataformes possibles: *Android Studio* i *Eclipse* amb ADT. He triat *Android Studio* principalment perquè és la plataforma oficial de *Google* pel desenvolupament d'aplicacions, ja que no tenia coneixements previs de cap de les dues.

A la tenda d'aplicacions d'*Android*, el *GooglePlay*, existeixen diferents aplicacions per la gestió de les reserves de pistes pels clubs de tennis i pàdel de les quals n'he agafat algunes per comparar-les amb la meua.

Cal destacar que tot i que totes tenen la capacitat de gestionar reserves amb funcions senzilles i d'altres amb grans possibilitats per gestionar diversos clubs al mateix temps no he trobat que cap de les principals també gestioni d'alguna manera els torneigs que es puguin fer als diferents clubs. A continuació en mostra alguna d'elles i s'explica una mica la seva descripció i les seves funcionalitats.

2.2.1. “Coge pista”

És una aplicació per gestionar les reserves de les pistes de pàdel d'urbanitzacions i complexos esportius on també es poden consultar les normes d'ús de les diferents urbanitzacions i complexos on hi ha les pistes.



Il·lustració 2: “Coge pista”. Font: [Externa](#)

2.2.2. “Club de Tennis Chamartín”

En aquesta aplicació els socis del club tenen la possibilitat de reservar i gestionar tota la seva informació, mantenint a més els tradicionals canals de reserva (online, telefònica o des de les pròpies instal·lacions).



Il·lustració 3: “Club de Tennis Chamartín”. Font: [Externa](#)

2.2.3. “Playtomic - Reserva pista de pádel y tenis”

Aquesta aplicació permet reservar pista de pádel i tennis en més de 500 clubs a l'instant, pagar el mateix preu que al club, mantenir les condicions especials dels socis de clubs.



Il·lustració 4: “Playtomic - Reserva pista de pádel y tennis”. Font: [Externa](#)

3. Anàlisis

En aquest capítol es presenta la fase d'anàlisis on es defineixen els requisits del projecte i es mostra una vista global de l'arquitectura pensada pel sistema. En el següent capítol es tindrà en compte aquest catàleg de requisits com a base pel disseny de tots els aspectes de l'aplicació. És per això que la fase d'anàlisis és molt important per dur a terme qualsevol producte de software. En aquest apartat és on s'han de crear les bases del projecte a partir de les quals es construirà tota la resta.

El procés a seguir es basa en primer lloc en una bona definició dels requisits i triar una metodologia de desenvolupament que s'adapti bé al projecte actual. L'arquitectura de l'aplicació tindrà el format de client-servidor, constituint el servidor de la base de dades, les *APIs* d'accés, el client i l'aplicació mòbil.

Aquest capítol és de vital importància per entendre el projecte. A continuació es desglossen la funcionalitat i les característiques del catàleg de requisits, tenint en compte tant els requisits funcionals com els no funcionals.

3.1. Catàleg de requisits

El catàleg de requisits és l'especificació del comportament que s'espera de qualsevol projecte de *software*. Estudiant altres aplicacions similars, s'han definit una sèrie de requisits que es consideren indispensables pel projecte. A continuació es mostra una enumeració i breu descripció dels requisits establerts pel disseny i desenvolupament de l'aplicació.

3.1.1. Requisits funcionals

Els requisits funcionals descriuen totes les interaccions que tindran els usuaris amb el *software*.

GESTIÓ D'USUARIS

S'exposen a continuació tots els requeriments associats a les múltiples accions que es podran realitzar com un usuari de l'aplicació sobre el seu propi compte.

1) Registre

1. L'aplicació ha de permetre a l'usuari introduir les seves dades en un formulari de registre on s'especificarà el sobrenom de l'usuari (“nickname”), el nom, el cognom, el correu electrònic, el telèfon, la contrasenya i el sexe de la persona.
2. El sistema s'encarregarà de validar les dades entrades per veure que totes compleixin les especificacions requerides.

3. El sistema mostrarà un missatge d'error si alguna de les dades és incorrecte o no compleix les condicions especificades del formulari de registre.
 4. En el cas que la validació sigui correcte, el sistema s'encarregarà de guardar les dades de l'usuari a la base de dades.
 5. El sistema enviarà un correu electrònic de confirmació al correu que ha introduït l'usuari al registre en cas que s'hagi efectuat correctament. Un cop l'usuari rebí el correu podrà activar el seu compte clicant en l'enllaç adjunt, en el missatge del correu.
 6. Si l'usuari no activa el seu compte, el sistema l'avisarà dient-li que ha d'activar abans el compte pel correu electrònic quan intenti iniciar sessió.
- 2) **Inici de sessió**
1. Per iniciar sessió l'usuari ha d'identificar-se pel seu sobrenom d'usuari ("nickname") i la seva contrasenya. En el cas d'haver iniciat sessió anteriorment amb un compte l'inici de sessió serà automàtic en tornar a iniciar l'aplicació amb aquell mateix compte.
 2. El sistema s'encarregarà de validar i permetre o denegar l'accés a l'aplicació i a les seves funcionalitats.
 3. El sistema mostrarà un missatge d'error en cas que la validació no sigui correcte.
 4. En el cas que la validació sigui correcte, es mostrarà un missatge de benvinguda a l'usuari i el redirigirà al menú principal on hi haurà totes les opcions disponibles per l'usuari.
- 3) **Tancament de sessió**
1. Qualsevol usuari de l'aplicació ha de poder finalitzar la seva sessió en l'aplicació mitjançant un botó que indiqui tancament de sessió.
 2. En el cas que un usuari cliqui el botó de tancament de sessió, el sistema mostrarà un missatge per assegurar-se que l'usuari vol tancar sessió, permeten acceptar o cancel·lar.
 3. En el cas que cancel·li el tancament de sessió no es farà res més i es mantindrà la sessió oberta.
 4. En el cas que s'accepti el tancament de sessió l'usuari serà redirigit al menú principal però només amb les opcions d'inici de sessió i registre disponibles.
- 4) **Sol·licitud de baixa**
1. L'usuari podrà sol·licitar donar-se de baixa en el sistema.
 2. El sistema mostrarà un missatge per assegurar-se que l'usuari vol donar-se de baixa, permeten acceptar o cancel·lar.
 3. En el cas que cancel·li no es farà res més i es mantindrà la sessió oberta i el compte de l'usuari intacte.
 4. En el cas que l'usuari cliqui acceptar es notificarà per correu electrònic que s'ha donat de baixa en l'aplicació. A més a més, es tancarà sessió automàticament redirigint al menú principal però només amb les opcions d'inici de sessió i registre disponibles.
- 5) **Compte d'usuari o editar usuari**
1. Ha d'existir una pantalla en el sistema que permeti a l'usuari consultar o modificar les dades del seu compte.
 2. Si l'usuari modifica alguna dada, el sistema validarà que les modificacions siguin possibles i en cas que no ho siguin es mostri un missatge d'error.
 3. En cas que la validació sigui correcte s'enviarà un correu de confirmació per poder confirmar els canvis.
 4. Si no es confirmen els canvis a través del correu, es mantindran les dades anteriors de l'usuari.
 5. Si es confirmen els canvis a través del correu, s'actualitzaran les dades a la base de dades.

APLICACIÓ

S'exposen a continuació tots els requeriments associats a les múltiples accions que es podran realitzar com un usuari de l'aplicació sobre les reserves, els torneigs i totes les altres opcions.

1) Menú principal

El menú principal mostrarà totes les opcions disponibles de l'aplicació amb botons grans amb nom juntament amb imatges intuïtives també clicables. Si un usuari ha iniciat sessió, tindrà totes les opcions disponibles, si no només tindrà disponibles l'inici de sessió i el registre. Al menú principal constarà de 9 botons amb les seves respectives imatges també clicables:

- a) **Inici de sessió:** Pantalla per iniciar sessió, tancar sessió, registrar-se o donar-se de baixa en l'aplicació.
- b) **Reservar pista:** Conjunt de pantalles per triar dia, pista i hores que l'usuari vulgui reservar.
- c) **Torneig de tennis:** Conjunt de pantalles per crear, registrar-se, consultar i participar en un torneig de tennis.
- d) **Torneig de pàdel:** Conjunt de pantalles per crear, registrar-se, consultar i participar en un torneig de pàdel.
- e) **Ubicació del club:** Pantalla per veure la ubicació del club.
- f) **Temps atmosfèric:** Pantalla per veure el temps atmosfèric, ja que les pistes del club són exteriors i interessa saber si plourà o no.

2) Menú lateral

El menú lateral mostrarà totes les opcions disponibles de l'aplicació amb botons amb nom juntament amb icones intuïtives també clicables. Si un usuari ha iniciat sessió, tindrà totes les opcions disponibles, si no només tindrà disponibles l'inici de sessió i el registre. El menú lateral constarà de 15 botons:

- a) **Menú principal:** Botó que permetrà anar al menú principal.
- b) **Inici de sessió:** Pantalla per iniciar sessió, tancar sessió, registrar-se o donar-se de baixa en l'aplicació.
- c) **Registre:** Pantalla per dur a terme el registre d'un usuari.
- d) **Editar usuari actual:** Pantalla que permetrà modificar qualsevol dada de l'usuari.
- e) **Menú d'administradors:** Menú on només poden accedir els administradors per gestionar l'aplicació.
- f) **Reservar pista:** Conjunt de pantalles per triar dia, pista i hores que l'usuari vulgui reservar.
- g) **Reserves actives:** Pantalla per veure les reserves actives de l'usuari actual en format llista.
- h) **Eliminar totes les reserves:** Botó que permetrà eliminar totes les reserves de l'usuari actual.
- i) **Torneig de tennis:** Conjunt de pantalles per crear, registrar-se, consultar i participar en un torneig de tennis.
- j) **Torneig de pàdel:** Conjunt de pantalles per crear, registrar-se, consultar i participar en un torneig de pàdel.
- k) **Historial de torneigs:** Pantalla on es podrà veure la informació dels torneigs antics i ja acabats.
- l) **Registre del temps atmosfèric:** Pantalla per veure el temps registrat gràcies a una aplicació externa.
- m) **Ubicació del club:** Pantalla per veure la ubicació del club.

- n) **Temps atmosfèric:** Pantalla per veure el temps atmosfèric, ja que les pistes del club són exteriors i interessa saber si plourà o no.
- o) **Contactar amb el club:** Pantalla que informará sobre el telèfon i el correu electrònic del club.

3) **Reserva de pista**

1. L'usuari podrà reservar hores en un dia concret i una pista concreta per poder jugar en una pista de tennis o de pàdel.
2. L'usuari podrà veure abans de reservar quines hores estan ocupades per altres usuaris i no les podrà seleccionar.
3. Hi ha d'haver la possibilitat de poder anular una reserva concreta i també la possibilitat de poder anular totes les reserves d'un mateix usuari encara que estiguin en dies diferents.
4. En cas que es dugui a terme una reserva correctament es guardarà a la base de dades i s'enviarà un correu electrònic (recordatori de la reserva) a l'usuari.

4) **Reserves actives**

1. Un usuari només hi podrà entrar si té reserves actives, és a dir, té reserves en el dia actual o més endavant. No mostrarà les reserves anteriors al dia actual.
2. Si té reserves actives, veurà les reserves en format llista separades per dies.

5) **Eliminar totes les reserves**

1. Botó que eliminarà totes les reserves actives de l'usuari actual independentment del dia.
2. Si no té cap reserva activa, apareixerà un avís i no passarà res.

6) **Torneig de tennis i torneig de pàdel**

1. L'usuari (participi o no al torneig) podrà entrar a veure el torneig existent, tant de tennis com de pàdel, per veure com van els partits i la gent que hi ha registrada en ells.
2. Un cop dintre d'un torneig l'usuari podrà registrar-se al torneig si encara no estan les inscripcions tancades, consultar els equips ja registrats i els seus integrants.
3. Un cop iniciat el torneig també podrà consultar els partits en llista i en format arbre i modificar un partit seu si participa en el torneig.

7) **Historial de torneigs**

1. Opció del menú principal per veure la informació dels torneigs ja acabats com les especificacions del torneig, la durada i el guanyador.
2. Si no hi ha cap torneig registrat, retornarà l'usuari al menú principal avisant amb un missatge del fet que no hi ha cap torneig registrat.

8) **Registre del temps atmosfèric**

1. Opció del menú lateral que permetrà veure en una llista els registres del temps atmosfèric que hagi fet el programa extern creat per mi.
2. Si no hi ha cap registre, retornarà l'usuari al menú principal avisant amb un missatge del fet que no hi ha cap registre atmosfèric disponible.

9) **Ubicació del club i temps atmosfèric:**

1. L'opció de la ubicació portarà a l'usuari al *Google Maps* amb les coordenades del club de tennis perquè pugui veure on és.
2. L'opció del temps atmosfèric portarà a l'usuari al *Meteocat* amb la població del club entrada perquè els usuaris puguin veure les previsions.

10) **Menú d'administradors**

El menú d'administradors tindrà tot un seguit d'opcions només disponibles pels administradors perquè puguin gestionar-ho tot des de la mateixa aplicació.

1. **Llista d'usuaris:** pantalla on els administradors podran veure tota la informació dels usuaris registrats a l'aplicació

2. **Llista d'administradors:** pantalla on es podrà consultar tota la informació dels usuaris registrats a l'aplicació i que també són administradors.
3. **Eliminar usuari:** possibilitat de poder eliminar un jugador guardant les seves dades claus en una llista negra per evitar que es pugui tornar a registrar sense que ho sàpiga un administrador.
4. **Totes les reserves:** pantalla on es podran veure totes les reserves actives independentment de l'usuari perquè els administradors puguin veure-ho ràpidament.
5. **Ocupar hores:** possibilitat dels administradors d'ocupar hores concretes que tindran prioritat sobre les reserves d'usuaris regulars i els hi anul·li la reserva avisant-los amb un correu electrònic.
6. **Ocupar dies:** opció per ocupar un dia sencer de manera que ningú hi pugui reservar i si algú ja hi havia reservat que els hi anul·li la reserva avisant-los amb un correu electrònic.
7. **Llista negra:** és una llista on es guardarà la informació clau dels usuaris (DNI, correu, "nick" i raó de l'expulsió) que hagin sigut expulsats per un administrador.
8. **Treure algú de la llista negra:** possibilitat de treure algú de la llista perquè pugui tornar a registrar-se sempre hi quan un administrador ho autoritzi.
9. **Missatge de difusió:** opció que només tindran els administradors d'enviar un missatge al correu electrònic de tots els usuaris (si es vol anunciar alguna notícia a tots els usuaris regular) i possibilitat d'enviar-ho només a tots els administradors (si es vol anunciar alguna cosa de manera interna).
10. **Omplir o buidar torneigs:** possibilitat que tindran els administradors d'omplir o buidar els torneigs que s'hagin creat però encara no estiguin actius perquè encara s'estan omplint.
11. **Modificar partits:** opció per modificar els partits de qualsevol torneig actiu.

11) Notificacions al correu electrònic

El sistema enviarà notificacions al correu electrònic en els següents casos:

- a) Quan l'usuari es registri a l'aplicació per activar el compte i confirmar el registre.
- b) Quan es faci una reserva per recordar el dia, la pista i les hores reservades.
- c) Quan un usuari es dona de baixa de l'aplicació.
- d) Quan un administrador elimini un usuari li arribarà un correu amb el motiu de l'expulsió.
- e) Quan un administrador enviï els missatges de difusió.
- f) Quan s'informi del temps als usuaris amb reserves.

3.1.2. Requisits no funcionals

Requisits complementaris o atributs de qualitat. Especifiquen criteris que jutgen operacions del sistema en lloc del seu comportament (requisits funcionals).

1) Documentació

1. Tot i que l'aplicació es farà pensant en el fet que sigui tan intuïtiva com sigui possible es durà a terme un manual d'usuari de l'aplicació per facilitar la seva utilització en cas de dubte.
2. La codificació del sistema haurà de ser clara i estar documentada de manera que qualsevol programador pugui agregar funcionalitats posteriorment procurant seguir els estàndards de la programació d'*Android*.

2) Seguretat

1. Per poder utilitzar l'aplicació t'hauràs d'autenticar.
2. Només t'hauràs d'autenticar un cop quan et registris.
3. Si l'usuari no tenca la seva sessió, es mantindrà oberta per futurs usos de l'aplicació.
4. Les contrasenyes dels usuaris seran xifrades.

3) **Sostenibilitat i portabilitat**

1. Disponibilitat en la gran majoria de dispositius *Android*.
2. L'aplicació *Android* estarà suportada per la majoria de versions possibles.
3. Necessària una connexió a internet sigui per *Wi-fi* o per tarifa de dades.

4) **Interfície i usabilitat**

1. L'aplicació ha de constar d'una interfície senzilla, atractiva i intuïtiva. De tal manera que el seu ús no suposi algun impediment o esforç a l'usuari a l'hora de fer ús de l'aplicació.
2. La introducció de dades ha d'estar estructurada procurant evitar errors.

5) **Rendiment**

1. S'esperen temps de resposta no superiors a 2 segons i menors en les consultes a la base de dades.
2. Tant els accessos a la base de dades com algun càlcul que es realitzi a l'aplicació no suposa massa càrrega pel dispositiu, pel que el rendiment serà òptim.

3.2. Metodologia

La metodologia és el conjunt de passos que s'han de dur a terme per poder resoldre tots els objectius de manera eficaç i ordenada per acabar obtenint una aplicació funcional. A cada un dels passos, es provaran els diferents prototips fins a arribar al producte final.

- 1) **Aprentatge de coneixements necessaris:** Buscar i investigar com es realitza el desenvolupament d'aplicacions *Android* per veure quina és la millor opció per dur-lo a terme. Per aquesta raó, també s'han d'aprendre les bases del llenguatge Java (ja que l'*Android* es programa en Java).
- 2) **Primer prototip de l'aplicació:** Aquest primer prototip ha d'estar format pels diferents coneixements apresos anteriorment per gestionar les pantalles i les seves interaccions com per exemple utilitzar *fragments*.
- 3) **Creació d'una base de dades local:** D'aquesta manera es podrà tenir un control més directe i ràpid de la base de dades per aconseguir obtenir l'objectiu del projecte de manera local. I un cop aconseguit gestionar bé el programa amb la base de dades local només s'ha de passar a un servidor remot.
- 4) **Creació d'un servei web local:** Igual que la base de dades el servidor primer el crearem localment per gestionar la base de dades. Per dur a terme aquest pas s'han d'aprendre coneixements del llenguatge PHP (per la gestió del servidor) i de JSON (per les transmissions d'informació).
- 5) **Aconseguir una comunicació estable:** Utilitzar la llibreria de Google *VOLLEY* per enviar peticions HTTPS POST (per fer modificacions a les taules) i GET (per obtenir informació de les taules), la qual ens permetrà comunicar-nos amb el nostre servidor web.
- 6) **Creació d'un servei web de domini públic:** Un cop testejat el correcte funcionament de l'aplicació i les comunicacions amb el servei web local passar-ho al servei web de domini públic amb la base de dades remota i aconseguir que funcioni correctament amb l'aplicació.

Cada pas estableix un punt important en l'evolució del treball perquè també es poden fer servir com a punts de restauració si apareix algun problema. Finalment, un cop testejada l'aplicació amb usuaris del club de tennis i pàdel de Navàs distribuir-la publicant-la a la *Play Store d'Android*.

3.3. Seguretat del projecte

Tenint en compte que la seguretat en qualsevol aplicació que gestioni informació és primordial per assegurar el seu correcte funcionament és important analitzar les diferents vulnerabilitats que poden aparèixer per poder controlar-les. S'ha de tenir en compte la seguretat de la informació, la seguretat de la comunicació i la seguretat a l'accedir a la base de dades.

3.3.1. Seguretat de la informació guardada

Per poder assegurar la integritat de la informació que s'emmagatzema, en aquest cas a la base de dades, es poden utilitzar diferents algorismes de xifratge per guardar amb tota seguretat la informació i evitar que se'n pugui fer un mal ús. A continuació es mostren els principals algorismes que s'utilitzen actualment per emmagatzemar la informació xifrada a les bases de dades. Finalment s'explicarà quina és la funció dels algorismes BLOWFISH i MD5 dins del treball.

1) MD5

En criptografia, MD5 (acrònim de Algorithme de Resum de Missatge 5) és un algorisme criptogràfic de reducció de 128 bits que permet l'obtenció d'una petita "signatura" o resum característic per a cada bloc de dades original que el representa. Un resum o *hash* de MD5 s'expressa habitualment com una seqüència de 32 dígits hexadecimals, que no permet reconstruir les dades originals, però permet saber si les dades originals han estat modificades. Com a part de l'estàndard establert per la RFC 1321 ha estat àmpliament utilitzat a una gran varietat d'aplicacions de seguretat i com a sistema de verificació de la integritat dels arxius.

Tot i haver estat considerat segur al principi, certes investigacions han revelat vulnerabilitats que fan qüestionable l'ús futur del MD5. A l'agost de 2004 van anunciar el descobriment de col·lisions de *hash* per MD5.

2) SHA

El SHA (Algorithme de *Hash* Segur) és una família de funcions *hash* de xifrat publicades per l'Institut Nacional de Normes i Tecnologia de EE. UU. La primera versió de l'algorisme es va crear el 1993 amb el nom de SHA, tot i que actualment se la coneix com SHA-0 per evitar confusions amb les versions posteriors. La segona versió del sistema, publicada amb el nom de SHA-1, va ser publicada dos anys més tard. Posteriorment s'han publicat SHA-2 a 2001 (formada per diverses funcions: SHA-224, SHA-256, SHA-384, i SHA-512) i la més recent, SHA-3. Aquesta última versió es caracteritza per ser la que més difereix de les seves predecessores.

El 23 de febrer de 2017, un equip format per Google i CWI Amsterdam va anunciar la primera col·lisió de SHA-1 posant en dubte l'algorisme.

3) BLOWFISH

Blowfish és un sistema de xifratge per blocs de clau dissenyat el 1993 per Bruce Schneier i s'inclou en un gran nombre de paquets de xifratge i productes d'enciptació. *Blowfish* proporciona un bon ritme d'enciptació en el programari i no s'ha trobat cap criptoanàlisi eficient fins a la data (2010). Tanmateix, en l'actualitat es presta més atenció al sistema AES.

Els trets notables del disseny inclouen Caixes-S dependents de la clau i una programació de claus altament complexa.

Blowfish té una longitud de bloc de 64 bits i una mida de clau variable des de 32 fins a 448 bits. És un xifratge per xarxa de *Feistel* de 16 rondes i fa servir claus dependents de les Caixes-S. És similar en estructura a CAST5, que fa servir caixes-S fixes.

Finalment, tenint en compte les possibles vulnerabilitats de MD5 i SHA1 (que són els estàndards per xifrar a MySQL) i després d'investigar altres possibilitats s'ha optat per utilitzar una llibreria de PHP que permet fer ús de l'algorisme *Bcrypt* (implementació amb PHP del algorisme BLOWFISH) per xifrar les contrasenyes emmagatzemades. Ja que les contrasenyes no poden ser visibles per cap persona que intercepti o gestioni la base de dades.

Per altra banda, pel codi d'activació utilitzat per activar el compte, s'utilitzarà l'algorisme MD5 i sortirà de xifrar el correu actual juntament amb un segell de la hora actual.



Il·lustració 5: Parts diferenciades del algoritme de xifratge Bcrypt. Font: [Externa](#)

ALGORITME DE XIFRATGE	TEST	REPOSTA
BCRYPT	12	\$2y\$10\$OpTSQAY0v0Po1i7sdmZ/MOx9Ik0lE8.6p3Va3qtEUe3QudCVNkkc2
MD5	12	c20ad4d76fe97759aa27a0c99bffa6710
SHA1	12	7b52009b64fd0a2a49e6d8a939753077792b0554

Taula 2: Exemple del xifratge amb els 3 possibles algoritmes

3.3.2. Seguretat en la comunicació de la informació

L'aplicació es comunica amb el servidor remot utilitzant peticions HTTPS per rebre i obtenir dades. D'aquesta manera les peticions porten un certificat per evitar els possibles atacs de "Man in the Middle" i es pot assegurar que ni la informació ni el certificat no s'hagin pogut modificar o falsificar pel camí, ja que s'ha aconseguit autenticar el certificat a l'hora de rebre les comunicacions.

Tenint en compte que la comunicació HTTP transmet en text pla sense xifrar, l'opció de poder utilitzar peticions HTTPS, que xifren la informació i la validen amb un certificat SSL dota les transmissions d'una seguretat molt superior.

3.3.3. Seguretat per accedir a la base de dades remota

Per poder accedir a la base de dades s'utilitzen fitxers PHP que gestionen les peticions i depenen dels paràmetres duen a terme les diferents consultes, modificacions i insercions a les diferents taules de la base de dades. Totes aquestes consultes han de ser tractades prèviament per evitar possibles injeccions SQL, que poden provocar greus danys a la base de dades i per aquesta raó a l'aplicació.

Primer de tot dir que els fitxers amb PHP utilitzats al servidor serveixen per gestionar totes les peticions que rep de l'aplicació. Des de l'aplicació s'utilitza la llibreria *VOLLEY* per generar les diferents peticions POST i GET i esperar una resposta donada pel servidor.

En rebre les peticions POST o GET els fitxers PHP miren quina petició s'està demanant concretament per gestionar la que toca mitjançant una connexió a la base de dades que permet realitzar les consultes, modificacions, insercions o eliminacions.

El llenguatge de programació PHP disposa de funcionalitats que permeten accedir a la base de dades amb *MySQL* de manera ràpida i senzilla. En les últimes versions de PHP fins a la 5 s'utilitzaven les funcions amb extensió "*MySQL*", però a partir d'aquesta versió es recomana utilitzar les extensions "*MySQLi*" perquè suposen una millora important davant de les versions anteriors, com per exemple:

- 1) Possibilitat de realitzar múltiples consultes utilitzant "*mysql_multi_query()*".
- 2) Interfície dual que suporta la programació orientada a objectes i la procedimental.
- 3) Permet la utilització de solucions directes a les injeccions SQL.
- 4) Suporta sentències SQL parametritzades.
- 5) Suporta transaccions ACID.
- 6) Millora de la seguretat.

Un cop analitzat com es comunica el servidor amb la base de dades s'ha de poder evitar que es puguin fer injeccions SQL a aquest. Les injeccions SQL són una via d'entrada per introduir codi intrús per poder realitzar consultes o modificacions no desitjades a la base de dades. Aquesta gran vulnerabilitat apareix si no es du a terme un correcte filtratge de les variables utilitzades per

una petició on hi podria haver camuflada una sentència SQL. Hi ha dos possibles procediments per evitar les injeccions SQL i no tenir problemes de seguretat:

1) Sentències parametritzades ("prepared statement")

Una sentència parametritzada és una característica utilitzada per executar repetidament les mateixes instruccions SQL (o similars) amb una alta eficiència.

Les declaracions preparades bàsicament funcionen així:

- 1) Preparar: es crea una plantilla d'estat de SQL i s'envia a la base de dades. Alguns valors no són especificats, anomenats paràmetres (etiquetats com "?").
Exemple: INSERT INTO MyGuests VALUES (?, ?, ?)
- 2) La base de dades analitza, compila i realitza l'optimització de consultes a la plantilla d'instruccions SQL i emmagatzema el resultat sense executar-lo.
- 3) Executar: en un moment posterior, l'aplicació vincula els valors als paràmetres i la base de dades executa la instrucció. L'aplicació pot executar la declaració tantes vegades com vulgui amb diferents valors.

En comparació amb l'execució d'instruccions SQL directament, les declaracions preparades tenen tres avantatges principals:

- Les instruccions preparades redueixen el temps d'anàlisi, ja que la preparació a la consulta es fa només una vegada (tot i que la instrucció s'executa diverses vegades).
- Els paràmetres vinculats minimitzen l'amplada de banda al servidor, ja que necessiten enviar només els paràmetres cada vegada, i no tota la consulta.
- Les instruccions preparades són molt útils contra les injeccions de SQL, ja que els valors dels paràmetres, que es transmeten posteriorment mitjançant un protocol diferent, no necessiten ser escapats correctament. Si la plantilla d'instrucció original no es deriva d'una entrada externa, la injecció de SQL no es pot produir.

Per diferenciar les diferents peticions dins d'un PHP i mirar també que no arriben amb valor nuls, s'utilitza la funció `isset()`, que s'utilitza en la recepció dels paràmetres per veure si les peticions arriben amb els camps que toquen i també per comprovar si la petició existeix (si no existeix, es descarta directament).

Funcions disponibles per crear sentències parametritzades són:

- **`MySQLi_prepare`**: Prepara la sentència SQL per a la seva posterior execució.
- **`MySQLi_bind_param`**: Relaciona i introdueix les variables que es reben com a paràmetres de la sentència parametritzada.
- **`MySQLi_execute`**: Executa la sentència parametritzada preparada prèviament.

Funcions disponibles per la comunicació amb la base de dades:

- **`MySQLi_connect`**: Executa una connexió directa amb el servidor on hi ha guardada la base de dades `MySQL`. Té com a paràmetres el nom del servidor, el nom d'usuari, la contrasenya per la connexió i el nom de la base de dades que es vol manipular, per si n'hi ha més d'una.
- **`MySQLi_query`**: Executa una sentència SQL (`SELECT`, `INSERT`, `UPDATE`, `REPLACE` o `DELETE`).
- **`MySQLi_fetch_array`**: Funció que retorna els resultats de les consultes i que permet accedir a les dades.
- **`MySQLi_affected_rows`**: Funció que retorna el nombre de files afectades en l'última consulta (`INSERT`, `UPDATE`, `REPLACE` o `DELETE`).

2) Transformació de les sentències (`MySQLi_real_escape_string`)

"`MySQLi_real_escape_string`" permet transformar les variables rebudes transformant els caràcters especials de SQL pel seu equivalent textual, de manera que, a l'hora d'interpretar el

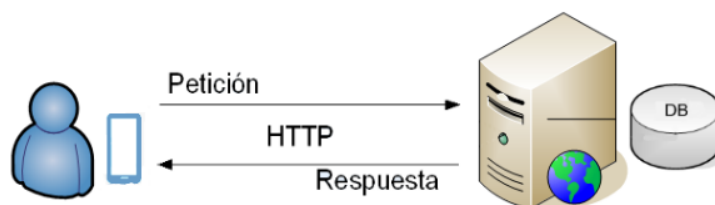
seu contingut veu les variables (prèviament transformades) com text normal i no com una possible sentència SQL.

4. Disseny

En aquest capítol es descriu tot el procés de l'aplicació del Club de Tennis i Pàdel. S'ha realitzat un disseny per aconseguir dur a terme tots els requisits descrits a l'apartat de [catàleg de requisits](#). El disseny proporciona una idea completa del software desenvolupat en el projecte. A més a més, es justifiquen totes les decisions preses pel seu posterior desenvolupament.

4.1. Arquitectura de l'aplicació

Com s'ha esmentat en l'apartat 3 ([Anàlisis](#)), l'arquitectura de l'aplicació es basa en el model client servidor. En el cas d'aquest projecte, i com es pot veure en la següent imatge, l'aplicació mòbil constituirà la part de client i la base de dades estarà en un servidor, el qual si accedirà mitjançant peticions HTTPS.



Il·lustració 6: Arquitectura del sistema

Per dur a terme aquest projecte era necessària una base de dades centralitzada i no local en el terminal, pel que la separació entre client i servidor ha d'estar ben diferenciada.

El client és qui inicia la sol·licitud, tenint un paper actiu en la comunicació i esperant la resposta del servidor. En el cas d'aquest projecte i com és comú en models d'aquest tipus, el client interactua amb l'usuari a través d'una interfície gràfica.

Per altra banda, el receptor de la sol·licitud, el servidor, espera que arribin les sol·licituds dels clients, exercint un paper passiu en la comunicació. Després de cada recepció processa i interactua amb la base de dades per acabar enviant les dades demanades cap al client.

Els avantatges que suposa aquesta separació són notables. Com per exemple avantatges de tipus organitzatiu a causa de la centralització de la gestió de la informació i separació de responsabilitats, el que clarifica i facilita el sistema. L'escalabilitat i encapsulació són altres dels avantatges més destacables d'aquest model. Permeten augmentar funcionalitats o recursos dels dos per separat.

Els desavantatges principals d'aquest model d'arquitectura són pocs però és important tenir-los en compte. Un exemple és que si el servidor deixa de funcionar per alguna raó, l'aplicació deixa de ser funcional per a tots els usuaris, ja que pràcticament totes les funcionalitats es comuniquen amb el servidor. A més a més, l'augment exagerat de tràfic de peticions, perquè per exemple molta gent comenci a utilitzar-la, també pot suposar un risc en el funcionament correcte del servidor.

4.2. Tecnologies utilitzades i eines de treball

En aquest apartat s'enumeren les diferents eines i tecnologies utilitzades durant el desenvolupament del projecte. Per cada una de les tecnologies i eines utilitzades es mostra la seva versió, la plataforma on s'ha utilitzat, la seva descripció, el seu logotip i el seu enllaç de descàrrega.

4.2.1. Android Studio

Versió: *Android Studio* 3.1.3. Build number: AI-173.4819257. *Android SDK Tools*: 26.1.1. *Android Platform API* 28 versió 4. Es pot descarregar en el següent [enllaç](#).

Plataforma: Windows 10 Home (64 bits) 2018 Microsoft Corporation.

Android Studio és un entorn de desenvolupament integrat oficial per la plataforma *Android*. Va ser anunciat el 16 de Maig de 2013 en la conferència de *Google I/O* i va reemplaçar al Eclipse com l'IDE oficial pel desenvolupament.

Està basat en el software IntelliJ IDEA i de JetBrains i ha sigut publicat de forma gratuïta a través de la llicència Apache 2.0. Està disponible per les plataformes Windows, macOS i GNU/Linux. Ha sigut dissenyat específicament pel desenvolupament d'*Android* i el seu llenguatge de programació és el Java.



Il·lustració 7: Logotip d'Android Studio. Font: [Externa](#)

Algunes de les seves característiques principals són:

a) Editor visual de pantalles

Editor capaç de poder crear dissenys complexos amb "ConstraintLayout" afegint restriccions de cada visualització a altres visualitzacions i directrius. A més a més, es pot visualitzar el disseny en qualsevol mida de la pantalla si seleccioneu una de diverses configuracions del dispositiu o simplement canviar la mida de la finestra de vista prèvia.

b) Analitzador APK

Amb l'analitzador es pot reduir la mida de l'aplicació d'*Android* inspeccionant el contingut del fitxer APK de l'aplicació, encara que no s'hagi creat amb *Android Studio*. Inspecciona el fitxer de manifest, els recursos i els fitxers DEX. A més a més, pot comparar dos APK per veure com la mida de l'aplicació ha canviat entre les versions de l'aplicació.

c) Emulador

Possibilitat de instal·lar i executar les aplicacions més ràpidament que amb un dispositiu físic i simular configuracions i funcions diferents, inclosa ARCore, la plataforma de *Google* per crear experiències de realitat augmentada.

d) Editor de codi intel·ligent

Permet escriure un codi millor, treballar més ràpid i ser més productiu amb un editor de codi intel·ligent que proporciona el suport de codi per als idiomes Kotlin, Java i C / C ++.

e) Sistema de compilació flexible

El sistema de compilació d'*Android Studio* permet personalitzar la compilació per generar diverses variants de compilació per als diferents dispositius d'un únic projecte.

f) Estadístiques de funcionament en temps real

Les eines de perfils integrades proporcionen estadístiques en temps real per a la CPU, la memòria i l'activitat de la xarxa de l'aplicació. D'aquesta manera es poden identificar els colls d'ampolla de rendiment gravant traces del mètode, inspeccionant les assignacions i veien les càrregues de xarxa entrants i sortints.

L'estructura bàsica del codi font d'un projecte *Android* es divideix en 3 parts diferenciades:

- 1) **Android Manifest:** dins de l'arxiu *AndroidManifest.xml* hi ha guardada tota la informació important de l'aplicació, la qual, utilitzarà el sistema *Android* per tal de poder executar-la. A més a més, descriu les diferents pantalles de l'aplicació que l'integren, declara els permisos per utilitzar qualsevol component de l'aplicació, utilitza les biblioteques vinculades, declara el rang de versions d'*Android* que suporta i descriu els paquets de Java que utilitza.
- 2) **Java:** hi ha els fitxers que contenen el codi font de Java que permet descriure les interaccions dels botons, *spinners*, editors de text, diàlegs, etc. A més a més també pot contenir la informació d'altres classes no associades a la interfície gràfica.
- 3) **Res:** si troben tots els fitxers que controlen tots els recursos, com per exemple els dissenys de pantalles (XML), les imatges, algunes cadenes de caràcters de la integració gràfica, etc. Per aquestes raons és on es troba tot el contingut gràfic de l'aplicació.

4.2.2. XAMPP

Versió: 7.2.9 / PHP 7.2.9. Suma de comprovacions bàsiques: md5, sha1. Es pot descarregar en el següent [enllaç](#).

Plataforma: Windows 10 Home (64 bits) 2018 Microsoft Corporation.

És un paquet de software lliure, que consisteix principalment en el sistema de gestió de base de dades *MySQL*, el servidor Apache i els intèrprets pel llenguatge de scripts PHP i Perl.



Il·lustració 8: Logotip de XAMPP. Font: [Externa](#)

L'ús d'aquest programa dins del projecte recau en veure com es comporta l'aplicació dins de un servidor local i la comunicació amb la base de dades abans d'utilitzar un servidor remot de domini públic.

4.2.3. Servidor HTTP Apache

Versió: Apache HTTP Server 2.4.34 (2018-07-16)

Plataforma: Windows 10 Home (64 bits) 2018 Microsoft Corporation.

El servidor HTTP Apache és un servidor web HTTP de codi obert. Apache presenta entre altres característiques, bases de dades d'autenticació i negociació de contingut. Pel que és una gran eina per poder gestionar les peticions HTTP que li arriben i poder donar una resposta al client que ha enviat la petició.



Il·lustració 9: Logotip d'Apache. Font: [Externa](#)

És el tipus servidor que crea Xampp per poder fer proves de manera local juntament amb la base de dades per veure com interacciona.

4.2.4. SQLyog

Versió: 13.0 de 64 bits.

Plataforma: Plataforma: Windows 10 Home (64 bits) 2018 Microsoft Corporation.

És un gestor de base de dades en *MySQL* sobre el sistema operatiu de Windows. S'utilitza principalment per gestionar la base de dades local creada a partir de Xampp. Es una eina de gestió de base de dades potent compacta i molt ràpida d'escriptori per gestionar les bases de dades *MySQL*. SQLyog és un administrador de bases de dades fàcil d'utilitzar per *MySQL* i MariaDB, que combina les funcions de *phpMyAdmin*, l'administrador de *MySQL*, les eines GUI

de *MySQL*, els clients de *MySQL* i altres interfícies *MySQL* en una única interfície intuïtiva. *SQLyog* ofereix una sofisticada eina de gestió de bases de dades, ofereix la manera més ràpida de comunicar-se amb el servidor *MySQL*, permet l'accés directe al servidor *MySQL* amb SSH i Tunneling HTTP, molt més fàcil de realitzar còpies de seguretat, transferència i migració amb un ús mínim del registre.



Il·lustració 10: Logotip de *SQLyog*. Font: [Externa](#)

Les característiques que té aquest gestor de base de dades són:

- Opcions de connectivitat avançades (HTTP / HTTP / SSH / SSL)
- Eines de migració avançades, dades d'importació i exportació
- Eines completes de gestió d'usuaris i diagnòstic
- Gestió de diverses connexions de base de dades
- Cerca intel·ligent de dades, visualització i comparació visual
- Taula principal de consulta amb interfície multicolor
- Editor de consultes amb diverses fitxes i editor de resultats
- Codi intel·ligent i generador de consultes
- Eina de sincronitzador, programador i informes

4.2.5. *MySQL*

Versió: 5.5.30-cll - *MySQL*

Plataforma: Windows 10 Home (64 bits) 2018 Microsoft Corporation.

MySQL és un sistema de gestió de base de dades relacional desenvolupat sota llicència dual i és considerat com la base de dades de codi obert més popular del món sobretot per entorns de desenvolupament web.



Il·lustració 11: Logotip de *MySQL*. [Externa](#)

Com a administrador de bases de dades relacional, *MySQL* emmagatzema les dades en forma de taules estructurades, amb camps, índexs, claus foranes i integritat referencial en l'actualització o esborrat en cascada. Utilitza el llenguatge SQL per a les diferents transaccions de dades.

Es pot instal·lar pràcticament en qualsevol sistema operatiu: Unix, Linux, Windows, MacOS; sol o acompanyat d'altres programes de l'entorn de programació com el servidor web Apache, PHP i *PhpMyAdmin*; en les distribucions WAMP, LAMP o XAMPP. Algunes de les característiques que posseeix són:

- Autenticació d'usuaris amb permisos específics per a certes bases de dades, lligades a les adreces IP d'origen.
- Gestió de memòria i la memòria cau per a una quantitat determinada de consultes o transaccions simultànies.
- Connectors per integració en ambients PHP, Perl, Python, ODBC.
- Replicació transaccional en línia de la base de dades, a una altra base paral·lela.
- Monitoratge d'usuaris, transaccions, ús de memòria i de processos.
- Ús de *triggers* o disparadors per actuar sobre creació, edició o esborrat de registres.
- Ús automàtic de valors numèrics automatitzats a les taules.
- Es poden crear vistes, procediments emmagatzemats i funcions.

- Posseeix funcions de revisió de l'acompliment de consultes i els seus índexs, com també dels paràmetres de memòria, la memòria cau, transaccions, etc. Podem trobar un conseller d'acompliment en *PhpMyAdmin*.
- Es pot administrar des de consola, *phpMyAdmin*, o des de programes externs com *MySQLfront* o *Sequel* (per a MacOS) mitjançant connexió pel port TCP 3306.
- Pot manejar milions de registres en una sola taula.
- Gestiona el bloqueig de taules per evitar conflictes de transaccions simultànies.
- Es poden importar o exportar dades o l'estructura mateixa de la base, les seves taules, índexs, etc.

4.2.6. 000webhost

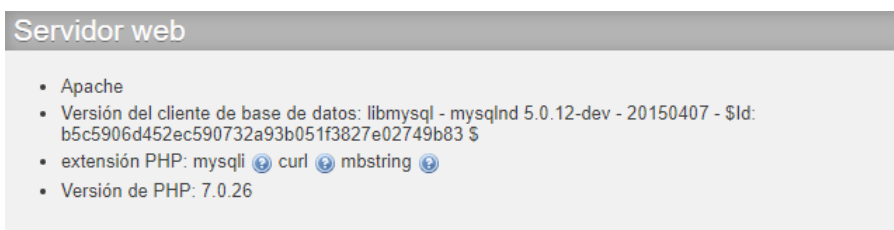
És un servei d'allotjament web gratuït que permet als usuaris publicar i gestionar llocs web de manera gratuïta. El servei web ofereix les següents prestacions de manera gratuïta:

- 1) 1000 MB d'espai en disc
- 2) 10 GB d'amplada de banda
- 3) 2 bases de dades *MySQL*
- 4) 2 llocs web
- 5) Creador de lloc web
- 6) *Hosting* del domini
- 7) 1 hora/dia de temps de manteniment (es pot triar per fer que sigui de matinada)



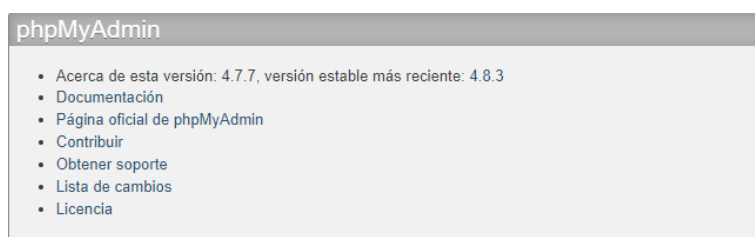
Il·lustració 12: Logotip de 000webhost. [Externa](#)

Permet utilitzar bases de dades *MySQL*, disposa d'un gestor d'arxius a través de la web, recuperació d'arxius, suport de FTP i les últimes versions de PHP (7.2). Té implementada la protecció *hotlink*, és a dir, evitar que altres serveis web puguin vincular arxius o serveis al teu servidor web. A més a més, permet protegir els directoris dels arxius del servidor web amb una autenticació d'usuari i contrasenya.



Il·lustració 13: Informació del servidor web

Tenint en compte la gestió de la base de dades es pot controlar mitjançant la interfície *phpMyAdmin* (versió 4.7.7) entre d'altres. La versió PHP que s'hi utilitza és la 7.0.26 i el servidor web és un servidor Apache 2.4.6 que utilitza OpenSSL com a protocol de seguretat per la comunicació de la xarxa.



Il·lustració 14: Informació del gestor *phpMyAdmin*

OpenSSL és una llibreria utilitzada per la majoria de serveis webs la qual conté una implementació de codi obert del protocol TLS i SSL. El servidor de la base de dades utilitza UNIX socket, que és un servidor Maria DB (versió 10.1.31) amb suport UTF8.

Servidor de base de datos
<ul style="list-style-type: none">• Servidor: Localhost via UNIX socket• Tipo de servidor: MariaDB• Versión del servidor: 10.1.31-MariaDB - MariaDB Server• Versión del protocolo: 10• Usuario: id6206731_xemnas@2a02:4780:bad:c0de::14• Conjunto de caracteres del servidor: UTF-8 Unicode (utf8)

Il·lustració 15: Informació del servidor de base de dades

4.3. Llenguatges de programació

S'han utilitzat 3 llenguatges de programació ben diferenciats per dur a terme les diferents parts del projecte: [Java](#), [PHP](#) i [Python](#).

4.3.1. Java

Versió: Java 8

Plataforma: *Android Studio*

És un llenguatge de programació de propòsit general, concurrent, orientat a objectes que va ser dissenyat específicament per tenir tan poques dependències d'implementació com fos possible. La seva intenció és permetre que els desenvolupadors d'aplicacions escriguin el programa una vegada i l'executin en qualsevol dispositiu. És el llenguatge de programació principal de l'*Android Studio*.

4.3.2. PHP

Versió: PHP 7.0.26

Plataforma: Servidor web

El llenguatge PHP (Hyper Text Preprocessor) és un llenguatge de programació de propòsit general de codi obert normalment dedicat al desenvolupament web i que pot ser incrustat en llenguatge HTML. Utilitzat també en servidors per gestionar peticions HTTP de qualsevol tipus, el que és perfecte per assolir el objectius d'aquets projecte.

Utilitzem aquest llenguatge per rebre les peticions dels clients, siguin POST o GET, i per connectar-nos amb la base de dades per aplicar canvis en ella o per llegir-la depenen de la petició rebuda.

En aquest cas concretament s'ha utilitzat una llibreria que s'anomena *PHPMailer* per poder enviar correus electrònics als diferents usuaris quan es necessita.

4.3.3. Python

Versió: Python 2.7.12

Plataforma: Ubuntu 16.04 LTS de 64 bits

Es tracta d'un llenguatge de programació multi paradigma, ja que suporta orientació a objectes, programació imperativa i, en menor mesura, programació funcional. És un llenguatge interpretat, usa tipus dinàmics i és multi plataforma.

És administrat per la *Python Software Foundation*. Posseeix una llicència de codi obert, anomenada *Python Programari Foundation License*, 3, que és compatible amb la Llicència Pública General de GNU a partir de la versió 2.1.1, i incompatible en certes versions anteriors.

4.4. Llibreries externes utilitzades

Per dur a terme gran part de les funcionalitats claus d'aquest projecte s'han utilitzat diferents llibreries. Algunes llibreries no es trobaven integrades dins de les dependències de l'*Android Studio*, del *PHP* o del *Python* i s'han hagut d'importar.

- *Android Studio*: [Volley](#).
- *PHP*: [PHPMailer](#), [Password Hash](#) i [MySQLi](#).
- *Python*: [Pyown](#), [Requests](#), [Time](#).

De cada un d'ells s'exposa a continuació una breu descripció, el logotip i l'enllaç de descàrrega.

4.4.1. Volley

Versió: *Volley* 1.1.0 disponible en aquest [GitHub](#).

És una llibreria de *Google* que permet utilitzar peticions HTTP a través de la xarxa per aplicacions d'*Android* de manera fàcil i ràpida.

Ofereix les següents característiques:

- Programació automàtica de sol·licituds de xarxa.
- Connexions de xarxa concurrents múltiples.
- Emmagatzematge en memòria cau transparent de memòria de disc i memòria amb coherència estàndard de memòria cau HTTP.
- Suport per a la prioritització de la sol·licitud.
- Sol·licitud d'anul·lació API. Es poden cancel·lar sol·licituds o es poden establir blocs o àmbits de sol·licituds de cancel·lació.
- Facilitat de personalització.
- Ordenació sòlida que fa que sigui fàcil omplir correctament la UI amb dades obtingudes de manera asíncrona des de la xarxa.
- Eines de depuració i traça.

Aquesta llibreria permet comunicar l'aplicació a través de la xarxa per poder realitzar peticions al servidor web per consultar, afegir o modificar informació de la base de dades. Perquè funcioni correctament caldrà donar permís de connexió a internet al *AndroidManifest*.

4.4.2. PHPMailer

Versió: *PHPMailer* 6.0.5 disponible en aquest [GitHub](#).

Molts desenvolupadors de *PHP* utilitzen el correu electrònic en el seu codi. L'única funció *PHP* que suporta aquesta és la funció "mail()". Tanmateix, no proporciona cap ajuda per fer ús de funcions populars, com ara correus electrònics i fitxers adjunts basats en HTML.

La funció de correu de *PHP* s'envia a través d'un servidor de correu local però normalment no inclou un servidor de correu local. La implementació SMTP integrada de *PHPMailer* permet l'enviament de correu electrònic a plataformes Windows sense un servidor de correu local.



Il·lustració 16: Logotip de PHPMailer. Font: [Externa](#)

Aquesta llibreria s'utilitza per diferents funcions de l'aplicació com per exemple la confirmació per correu electrònic de l'activació del compte, dels canvis fets en un compte o dels avisos de les reserves.

4.4.3. Password Hash

Llibreria disponible a partir de la versió 5.3 de PHP i que ofereix suport per molts algoritmes de *hashing* com per exemple MD5, SHA-1, SHA256, SHA-512, BCrypt, etc. Dins del projecte s'utilitzen dues funcions d'aquesta llibreria, una per xifrar (`password_hash`) les contrasenyes i l'altre per verificar-les (`password_verify`) sense veure el seu autèntic valor per raons de seguretat.

- 1) **`password_hash`**: aquesta funció té 2 variables, una és la contrasenya normal i l'altre és el tipus d'algoritme que s'utilitzarà per xifrar la contrasenya. Els dos algoritmes més importants per aquest tipus de xifratge són 2 (`PASSWORD_DEFAULT` i `PASSWORD_BCRYPT`).

El `PASSWORD_DEFAULT` és més segur perquè s'actualitza cada cop que apareix un algoritme més fort tot i que això porta a darrere el problema que després el resultat canvia de mida depenen del algoritme, el que pot resultar un problema a l'hora de guardar-ho a la base de dades.

En canvi, per altra banda, el `PASSWORD_BCRYPT` utilitza l'algoritme BLOWFISH per generar el xifratge que té sempre com a resultat una cadena de 60 caràcters.

- 2) **`password_verify`**: aquesta funció també té 2 variables, en una s'ha d'introduir el valor que es vol comparar i en l'altre la informació xifrada. A continuació retorna un `True` o un `False` depenen de si la informació, en aquest cas la contrasenya, té el mateix valor que la informació xifrada o no.

4.4.4. MySQLi

Aquesta funció permet accedir a la base de dades *MySQL* i utilitzar totes les funcionalitats que proporciona a partir de la versió 5 del PHP en endavant. A més a més, permet utilitzar un conjunt de mètodes per gestionar la base de dades *MySQL*. Les funcions més utilitzades en els fitxers PHP del servidor per comunicar-se amb la base de dades són *MySQLi_connect*, *MySQLi_query*, *MySQLi_fetch_array*, *MySQLi_affected_rows*, etc. Aquestes funcions estan explicades amb més detall a l'apartat de la [seguretat per accedir a la base de dades](#).

4.4.5. PyOWM

Llibreria de *python* que serveix per crear un client per connectar el programa a la pàgina de OpenWeatherMap i poder consultar el temps en un lloc concret (a la pista de tennis i pàdel de Navàs en aquest cas). Llibreria disponible en aquest [GitHub](#).



II-lustració 17: Logotip de la llibreria pyown. Font: [Externa](#)

Com que la pàgina web de OpenWeatherMap necessita una clau API de validació per permetre respostes, PyOWN no funciona si no se n'utilitza una. Aquesta clau està disponible tant per via de subscripció gratis amb algunes limitacions com per via de subscripció amb pagament sense limitacions. En aquest cas s'ha utilitzat el client de subscripció gratis que una de les limitacions que té és que no es poden fer més de 60 consultes en menys d'un minut, és a dir, una consulta per segon. En aquest cas no hi ha problema, ja que es fa una consulta cada hora. Per aconseguir una clau de subscripció gratis només s'ha d'entrar a la [web de OpenWeather Map](#) i registrar-se.



II-lustració 18: Logotip de OpenWeatherMap. Font: [Externa](#)

4.4.6. Requests

Versió: 2.19.1 disponible dins d'aquesta [pàgina web](#).

És una llibreria de *python* que permet fer peticions HTTP i que en aquest cas s'utilitza dins del programa de *python* per enviar peticions POST cap al servidor web amb la informació del temps atmosfèric a la pista cada hora.

4.4.7. Time

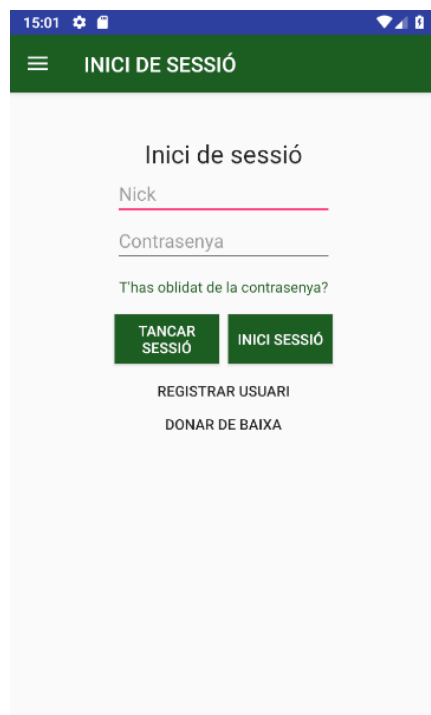
Versió: 2.5 amb la documentació disponible [aquí](#).

Llibreria oficial de *python* per poder utilitzar diferents funcions relacionades amb el temps. En aquest cas s'ha d'utilitzar per poder mirar l'hora del dia i fer una consulta en conseqüència o no.

4.5. Disseny del software de l'aplicació *Android*

4.5.1. Disseny de la interfície gràfica

La interfície s'ha dissenyat amb el propòsit que sigui intuïtiva i simple, però a la vegada potent i capaç de cobrir tots els requisits enumerats anteriorment en l'apartat de l'[anàlisi](#). Els usuaris podran navegar per ella sense problemes i amb poc ús entendre com funcionen totes les funcionalitats disponibles. Per veure amb més detall totes les funcionalitats i totes les pantalles disponibles de l'aplicació es poden consultar a l'[Annex. Manual d'ús de l'aplicació](#).



Il·lustració 19: Pantalla d'inici de sessió

El disseny de la interfície gràfica de totes les pantalles de l'aplicació està creat i pensat per a dispositius amb una resolució de 1080 x 1920 (FULL HD): 420dpi. Si la pantalla és més gran o té una resolució més gran podrien haver-hi parts de les pantalles sense components però totes les funcionalitats continuarien sent disponibles, ja que algunes no escalen juntament amb la pantalla (com els botons) però d'altres sí (com les llistes). Per altra banda, si la pantalla és més petita, s'ha tingut en compte afegint *scrollwiews* tant horitzontals com verticals a les pantalles que es necessiti. D'aquesta manera només movent la pantalla amb el contacte del dit es podran arribar a veure i a poder utilitzar totes les funcionalitats de cada pantalla.

4.5.2. Classes Java de l'aplicació *Android*

A continuació es mostren totes les classes Java que s'han creat per fer funcionar l'aplicació. Dividirem les classes Java en tres grups:

1. Classes Java sobre la interfície gràfica

Cada pantalla gràfica de l'aplicació té assignada una classe Java que permet realitzar totes les funcionalitats necessàries de cada pantalla. Les pantalles gràfiques estan connectades a les seves classes de Java utilitzant connectors que permeten interactuar entre les classes Java i els diferents elements gràfics com els botons, el quadres de text, etc.

Totes les diferents pantalles, que poden ser activitats o fragments, tenen associades una pantalla gràfica en format XML que conté la informació sobre els diferents elements gràfics sobre els que interactua un usuari final. A continuació es mostra una taula amb totes les classes que s'han utilitzat en la interfície gràfica.

Pantalla	Classe (.java)	Layout (.xml)
Buidar Torneig	buidarT	fragment_buidar_t
Treure "ban"	desbanejar_usuari	fragment_desbanejar_usuari
Eliminar Usuari	eliminar_usuari	fragment_eliminar_usuari
Llista administradors	llista_admins	fragment_llista_admins
Llista negra	llista_banejats	fragment_llista_banejats
Menú administradors	menu_admins	fragment_menu_admins
Missatge difusió	missatge_difusio	fragment_missatge_difusio
Modificar partit	modificar_partit	fragment_modificar_partit
Ocupar dies	ocupar_dies	fragment_ocupar_dies
Ocupar hores 1	ocupar_hores1	fragment_ocupar_hores1
Ocupar hores 2	ocupar_hores2	fragment_ocupar_hores2
Ocupar / buidar torneig	OcuparBuidarTorneig	fragment_ocupar_buidar_torneig
Ocupar torneig	ocuparT_Admin	fragment_ocupar_t_admin
Totes les reserves	TotesReserves	fragment_totes_reserves
Editar usuari	EditUser	fragment_edit_user
Iniciar sessió	Login	fragment_login
Registre	Register	fragment_register
Reserves usuari	ReservesUsuari	fragment_reserves_usuari
Historial torneigs	Historial_T	fragment_historial_t
Pantalla principal	MainActivity	activity_main
Menú principal	MenuP	fragment_menu_p
Torneig de pàdel	padelT	fragment_padel_t
Arbre 8 pàdel	padelTarbre8	fragment_padel_tarbre8
Arbre 16 pàdel	padelTarbre16	fragment_padel_tarbre16
Arbre 32 pàdel	padelTarbre32	fragment_padel_tarbre32
Crear torneig pàdel	padelTcrearT	fragment_padel_tcrear_t
Pàgina principal torneig pàdel	padelTprincipal	fragment_padel_tprincipal
Registre de pàdel	padelTregistre	fragment_padel_tregistre
Taula de partits de pàdel	padelTtaula	fragment_padel_ttaula
Calendari de reserves	Calendari	fragment_calendari
Horari de reserves	Horari	fragment_horari
Torneig de tennis	tennisT	fragment_tennis_t
Arbre 8 tennis	tennisTarbre8	fragment_tennis_tarbre8
Arbre 16 tennis	tennisTarbre16	fragment_tennis_tarbre16
Arbre 32 tennis	tennisTarbre32	fragment_tennis_tarbre32
Crear torneig tennis	tennisTcrearT	fragment_tennis_tcrear_t

Pàgina principal torneig tennis	tennisTprincipal	fragment_tennis_tprincipal
Registre de tennis	tennisTregistre	fragment_tennis_tregistre
Taula de partits de tennis	tennisTtaula	fragment_tennis_ttaula
Registre del temps atmosfèric	registre_temps	fragment_registre_temps
Temps atmosfèric	temps	fragment_temps
Ubicació del club	Ubicacio	fragment_ubicacio

Taula 3: Classes Java sobre la interfície gràfica

2. Classes Java per emmagatzemar informació

Són classes públiques encarregades de guardar la informació d'un element real com per exemple un equip o la informació d'un torneig. En aquest cas s'han utilitzat 5 classes Java diferents:

- **Equip individual:** classe pública utilitzada en la llista d'equips individual registrats a qualsevol dels torneigs. Serveix per guardar tota la informació d'un equip individual i poder obtenir-la més tard. A més a més, també serveix per emmagatzemar els valor obtinguts del servidor de manera unificada.

Funció	GET	SET
Nom equip	getNamet()	setNamet(String namet)
Nick	getNick()	setNick(String nick)
Nom jugador	getName()	setName(String name)
Correu electrònic	getEmail()	setEmail(String email)
Telèfon	getPhone()	setPhone(String phone)
Dades de la imatge	getDadesI()	setDadesI(String dadesI)
Bitmap de la imatge	getImatge()	setImatge(Bitmap imatge)

Taula 4: Funcions de la classe equip individual

- **Equip parella:** classe pública utilitzada en la llista d'equips per parelles registrats a qualsevol torneig per mostrar la informació de cada equip. En aquest cas, com que hi ha dos participants a cada equip es diferencien amb el número 1 i 2.

Funció	GET	SET
Nom equip	getNametP()	setNameP(String namet)
Nick 1	getNick1()	setNick1(String nick1)
Nom jugador 1	getName1()	setName1(String name1)
Correu electrònic 1	getEmail1()	setEmail1(String email1)
Telèfon 1	getPhone1()	setPhone1(String phone1)
Dades de la imatge 1	getDadesI1()	setDadesI1(String dadesI1)
Bitmap de la imatge 1	getImatge1()	setImatge1(Bitmap imatge1)
Nick 2	getNick2()	setNick2(String nick2)
Nom jugador 2	getName2()	setName2(String name2)
Correu electrònic 2	getEmail2()	setEmail2(String email2)
Telèfon 2	getPhone2()	setPhone2(String phone2)
Dades de la imatge 2	getDadesI2()	setDadesI2(String dadesI2)
Bitmap de la imatge 2	getImatge2()	setImatge2(Bitmap imatge2)

Taula 5: Funcions de la classe equip parella

- **Partit:** classe pública utilitzada per modificar els partits amb més facilitat, ja que cada partit té diferents característiques variables que es poden controlar millor si estan unificades.

Funció	GET	SET
Equip 1	getPlayer1()	setPlayer1(String player1)
Equip 2	getPlayer2()	setPlayer2(String player2)
Resultat	getResult()	setResult(String result)
Equip guanyador	getWinner()	setWinner(String Winner)
Equip perdedor	getLoser()	setLoser(String loser)

Taula 6: Funcions de la classe partit

- **Torneig:** classe pública per poder emmagatzemar i consultar després amb més facilitat la informació de cada torneig de manera separada i unificada.

Funció	GET	SET
Nom torneig	getName()	setName(String name)
Actiu	getActive()	setActive(String active)
Tipus	getType()	setType(String type)
Nivell	getLevel()	setLevel(String level)
Número jugadors	getPlayers()	setPlayers(String players)
Individual / Parelles	getPI()	setPI(String pi)
Control	getControl()	setControl(int control)
Guanyador	getWinner()	setWinner(String winner)

Taula 7: Funcions de la classe torneig

- **Usuari:** classe pública capaç d'encapsular tota la informació d'un usuari per poder consultar-la i modificar-la amb més facilitat.

Funció	GET	SET
Nick	getNick()	setNick(String nick)
Nom	getName()	setName(String name)
Cognom	getSurname()	setSurname(String surname)
Correu electrònic	getEmail()	setEmail(String email)
Telèfon	getPhone()	setPhone(String phone)
DNI	getDni()	setDni(String dni)
Sexe	getSexe()	setSexe(String sexe)
Dades de la imatge	getDadesI()	setDadesI(String dades I)
Bitmap de la imatge	getImatge()	setImatge(Bitmap imatge)

Taula 8: Funcions de la classe usuari

3. Classes Java per la creació de “Recycler View”

Aquestes classes permeten crear llistes amb diferents informacions i utilitzant algunes de les classes anteriors d'emmagatzematge per poder posar la informació en forma de llista.

- **EquipsIAdapter:** classe pública que actua com un adaptador per poder generar una llista amb la informació dels equips individuals. La llista interna que genera per mostrar la informació és del tipus Equip Individual (explicat a l'apartat anterior).
- **EquipsPAdapter:** classe pública que actua com un adaptador per poder generar una llista amb la informació dels equips per parelles. La llista interna que genera per mostrar la informació és del tipus Equip Parella (explicat a l'apartat anterior).
- **UsuariAdapter:** classe pública que actua com un adaptador per poder generar una llista amb la informació dels usuaris. La llista interna que genera per mostrar la informació és del tipus Usuari (explicat a l'apartat anterior).

4. Classes Java de comunicació

Aquesta classe permet millorar la comunicació de l'aplicació amb el servidor.

- **Singleton Volley:** a l' utilitzar la llibreria *Volley* amb moltes peticions diferents la documentació oficial recomana utilitzar un *Singleton* per encapsular totes les peticions en una sola llista. Per aquesta raó, com que aquesta aplicació utilitza moltes peticions és molt més eficient utilitzar una classe *Singleton*. També és important utilitzar-la, ja que les cues que es generen en enviar una petició tenen un cicle de vida limitat, depenen del context, en canvi així el cicle de vida és el de la mateixa aplicació.

4.6. Disseny de la base de dades

En aquest apartat es tracten els aspectes del disseny relacionats amb la base de dades del servidor. Principalment aquest apartat es divideix en el disseny del model entitat-relació i el detall de les taules creades per aquest projecte.

4.6.1. Disseny de les taules a la base de dades

Totes les taules d'aquesta base de dades han sigut creades amb el motor d'emmagatzematge "InnoDB". Que és un motor d'emmagatzematge molt útil per gestionar diferents operacions dins d'una mateixa base de dades sense saturar les taules.

"InnoDB" és el motor d'emmagatzematge per defecte actualment utilitzat pel gestor de base de dades "MySQL". El principal avantatge respecte altres és que proporciona transaccions que compleixen amb "ACID" (atomicitat, consistència, aïllament i resultats definitius) i també suporta claus foranes (per mantenir la integritat referencial en tot moment).

A continuació es poden observar 8 raons per les quals s'ha triat el motor d'emmagatzematge "InnoDB" davant d'altres motors com per exemple "MyISAM" (que és l'antic motor per defecte de "MySQL"):

1. **Suport de claus foranes:** aquest motor d'emmagatzematge permet relacionar taules de forma implícita en la base de dades. Això provoca que la mateixa base de dades s'encarregui d'eliminar inconsistències en les dades de les taules. De manera que podem assegurar que si es borra per exemple un producte també es borra automàticament la categoria associada a aquest. En canvi, amb "MyISAM" s'haurien de "suposar" les claus foranes i no podríem obtenir registres vàlids, ja que no s'estaria controlant. Per aquesta raó amb "MyISAM" podríem esborrar un producte i a la taula de la seva categoria hi quedarien registres orfes.
2. **Control de concurrència d'alt nivell:** utilitzant "MyISAM" quan una transacció modifica un registre d'una taula, la taula sencera queda bloquejada durant el temps que necessita per realitzar la modificació. Qualsevol altra transacció que es realitzi durant aquell període de bloqueig haurà d'esperar i això pot provocar coll de botella amb molta facilitat. "InnoDB", en canvi, proporciona un sistema de bloqueig a nivell de fila, el que significa que solament la fila que està sent modificada queda bloquejada. Altres transaccions intentant de modificar la mateixa taula podrien funcionar sempre i que modifiquessin altres registres. Modificar una taula de forma concurrent és altament probable, en canvi, modificar un mateix registre simultàniament és molt menys comú, el que dificulta i fa pràcticament impossible que es puguin produir colls de botella.
3. **Baix índex de taules corruptes:** Això és degut en part per la raó anterior, ja que a causa del mal control de concurrència de "MyISAM", és molt habitual trobar-se amb taules corruptes si una transacció falla. A "InnoDB" es compleix la norma que qualsevol transacció portarà a la base de dades d'un estat vàlid a un altre estat vàlid. A més disposa d'un registre de transaccions, el que fa que si una transacció falla, es torna a executar l'última de la llista.
4. **Índexs clusteritzats:** les dades s'emmagatzemen físicament per ordre (alfabètic o numèric depenen del tipus de la clau primària) del valor de la clau primària. Consultes que filtrin únicament pel valor de la clau primària seran extremadament eficients i ràpides, perquè en saber el valor que busquem, "InnoDB" amb un parell de senzilles operacions ja pot saber on es troba la dada.
5. **Millors opcions de replicació:** la replicació consisteix a copiar les dades de MySQL en diversos servidors per repartir la feina a l'hora de realitzar consultes. En utilitzar la replicació és fonamental que tots els nodes (servidors amb còpia de la base de dades) tinguin una còpia consistent de les dades, perquè si no cada node podria retornar resultats diferents. "InnoDB" pot germanitzar la còpia consistent gràcies al seu control de transaccions i a l'avançat sistema de bloqueig.
6. **Motor ACID:** com ja s'ha explicat anteriorment a la introducció d'aquest punt "InnoDB" proporciona transaccions que compleixen amb "ACID" (atomicitat, consistència,

aïllament i resultats definitius). D'aquesta manera la base de dades sempre va d'un estat vàlid a un altre vàlid.

7. **Taules sense límit de mida:** tot i que sempre hi ha les limitacions físiques de mida per guardar dades amb "MyISAM" cada taula es guarda en un arxiu separat i si la taula supera els 2 GB, és possible que el sistema d'arxius no sigui capaç d'utilitzar-lo. Per altra banda, "InnoDB" s'encarrega que les taules no tinguin límit de mida comprimint la informació de les files i dividint les dades en fitxers més petits.
8. **Índexs de *hash* adaptables:** si un índex es pot guardar en la memòria RAM i una taula és consultada freqüentment, és possible que "InnoDB" creï un índex *hash* en memòria. El que fa és replicar l'índex que es guarda en disc a la memòria RAM, de manera que els accessos tinguin una velocitat molt superior. És un sistema complex que pot augmentar el rendiment i velocitat de les consultes significativament.

També s'ha tingut en compte que totes les taules tinguin suport d'UTF-8 per tal de no tenir problemes a l'hora de guardar lletres amb accents o símbols estranys com la "ñ".

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
admins	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8_general_ci	16 KB	-
ban	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8_general_ci	48 KB	-
h_tournaments	Examinar Estructura Buscar Insertar Vaciar Eliminar	6	InnoDB	utf8_general_ci	16 KB	-
partit	Examinar Estructura Buscar Insertar Vaciar Eliminar	6	InnoDB	utf8_general_ci	80 KB	-
pertinencia	Examinar Estructura Buscar Insertar Vaciar Eliminar	16	InnoDB	utf8_general_ci	32 KB	-
registre	Examinar Estructura Buscar Insertar Vaciar Eliminar	8	InnoDB	utf8_general_ci	32 KB	-
reserva	Examinar Estructura Buscar Insertar Vaciar Eliminar	162	InnoDB	utf8_general_ci	32 KB	-
team	Examinar Estructura Buscar Insertar Vaciar Eliminar	8	InnoDB	utf8_general_ci	16 KB	-
temps_registre	Examinar Estructura Buscar Insertar Vaciar Eliminar	48	InnoDB	utf8_general_ci	16 KB	-
tournament	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8_general_ci	16 KB	-
user	Examinar Estructura Buscar Insertar Vaciar Eliminar	16	InnoDB	utf8_general_ci	6.6 MB	-
11 tablas	Número de filas	276	InnoDB	utf8_general_ci	6.9 MB	0 B

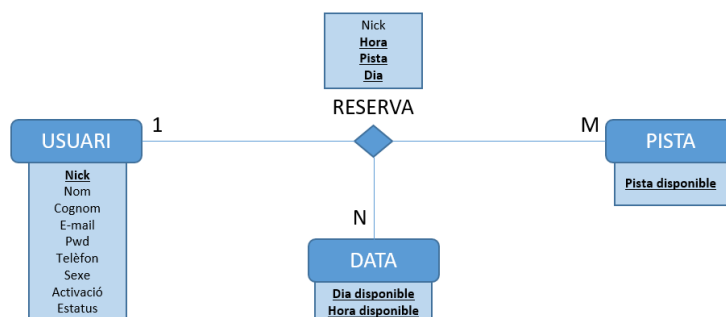
II·lustració 20: Vista global de totes les taules de la base de dades

4.6.2. Model entitat-relació

El model entitat-relació ha sigut dividit entre les taules relacions encarregades de gestionar les reserves de pistes i les taules relacionals encarregades de gestionar els torneigs perquè sigui més entenedor. Tot i això els dos esquemes tenen un nexce d'unió que és la taula d'usuaris pel que en realitat és un únic model d'entitat-relació més gran i s'ha dividit per facilitar l'explicació i per estructurar-ho millor.

1. Gestió de les reserves de pistes

Per dur a terme aquesta gestió s'han creat 4 taules a la base de dades, 3 de principals (usuari, data, pista) i una que apareix de la relació ternària 1:N:M de les altres 3 (reserva) tal com es pot veure en el següent esquema.



II·lustració 21: Esquema entitat-relació de la gestió de reserves

Observant l'esquema entitat-relació de la gestió de reserves es pot comprovar que la reserva té com a clau primària la unió del dia, la pista i l'hora, que és l'orde en el qual es fa una reserva. Primer es tria el dia en el calendari (primera part de la clau primària), després es tria la pista en

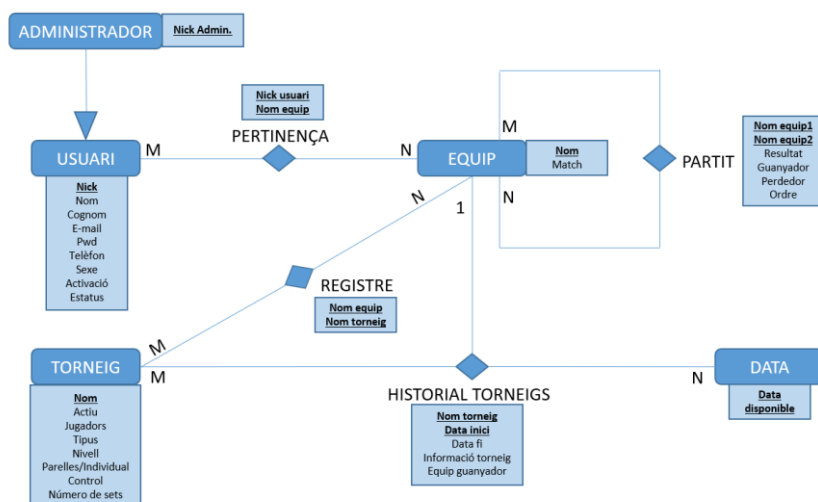
la següent pantalla (segona part de la clau primària) i finalment es trien les hores (tercera part de la clau primària). A més a més, com que la reserva només es pot fer si has iniciat sessió amb un usuari existent, la clau forana del “nick” que relaciona la taula usuari és el “nick” de l'usuari actual que està fent la reserva en aquell moment.

2. Gestió dels torneigs

Per dur a terme la gestió dels torneigs hem hagut de crear 9 taules a la base de dades, 5 de principals (administrador, usuari, torneig, equip, data) i 3 que apareixen de les relacions entre les principals (pertinença, registre, partit).

- Pertinença:** apareix a partir de la relació entre la taula d'usuari i la taula d'equip. És una relació M:N ja que un usuari pot ser de diversos equips al mateix temps i un equip pot tenir més d'un usuari diferent al mateix temps.
- Registre:** apareix a partir de la relació entre la taula d'equip i la taula de torneig. És una relació M:N ja que en un torneig hi pot haver més d'un equip i un equip pot estar a més d'un torneig diferent.
- Partit:** apareix a partir de la relació entre la taula equip i la taula equip una altra vegada, ja que un partit es compon de 2 equips que juguen entre ells. És una relació M:N ja que en un equip pot jugar contra més d'un equip diferent.

Totes les taules nomenades i les seves interrelacions es poden veure en el següent esquema.



Il·lustració 22: Esquema entitat-relació de la gestió de torneigs

Observat l'esquema entitat-relació de la gestió de torneigs es pot comprovar que la taula de pertinença serveix per emmagatzemar la relació entre els usuaris i els seus equips fent així possible que hi puguin haver equips de tants usuaris com es vulguin tot i que en el cas d'aquest projecte només seran equips d'un o de dos usuaris. Ja que tal com s'ha explicat anteriorment els torneigs de tennis només poden ser individuals o per parelles i els torneigs de pàdel només poden ser per parelles.

Els partits de la taula de partits es generen tenint en compte el número de cada equip per poder fer un emparellament aleatori i sense repeticions com explicaré més endavant en el disseny de torneigs. Tenint en compte la relació no es poden repetir partits però si que es poden repetir equips en diferents partits el que va perfecte per poder fer un torneig, ja que els equips que guanyin hauran de tornar a jugar amb altres equips però mai amb un equip el qual ja hagi perdut, ja que aquest ja ha estat eliminat.

3. Altres gestions

Hi ha 3 taules que serveixen per emmagatzemar informació que no estan vinculades directament amb les altres. Aquestes 3 taules independents són:

- a) **Historial de torneigs:** no està vinculada amb cap altre taula, ja que guarda la informació de l'últim torneig just abans d'eliminar-lo de la seva taula.
- b) **Llista negra:** no està vincula amb cap altre taula, ja que guarda la informació d'un usuari just abans d'eliminar-lo de la seva taula.
- c) **Registre del temps atmosfèric:** no està vinculada a altres taules, ja que guarda la informació del temps atmosfèric que arriba directament de les peticions enviades pel programa de *pyhton*.



II-lustració 23: Taules per altres gestions

4.6.3. Creació i format de les taules

La creació i format de les taules s'ha dut a terme mitjançant *MySQL* com ja s'ha mencionat anteriorment. A continuació veurem la creació de cada una de les taules i el seu format juntament amb l'explicació de per què s'han dissenyat així.

1. Taula d'usuaris

És la taula encarregada de guardar tots els usuaris registrats en l'aplicació i tota la seva informació per poder-la utilitzar quan sigui necessari.

Per crear aquesta taula s'han tingut en compte totes les dades que interessava guardar a la base de dades referents a cada usuari. En les primeres columnes es guarda la informació característica de cada usuari com el "nick", el nom, el cognom, el correu electrònic, la contrasenya, el telèfon, i el sexe de l'usuari. Les 2 columnes finals anomenades activació i estatus serveixen per comprovar l'activació de cada compte d'usuari. L'activació s'explicarà amb més detall a l'apartat del [disseny de la seguretat del client](#).

id6206731_tpclub	user
nick	: varchar(20)
name	: varchar(20)
surname	: varchar(20)
email	: varchar(30)
pwd	: varchar(100)
phone	: varchar(20)
sexe	: varchar(20)
activation	: varchar(300)
status	: enum('0','1')
dni	: varchar(20)
imatge	: longblob

II-lustració 24: Informació de la taula d'usuaris

Tal com es pot observar a la imatge de la informació de la taula d'usuaris la clau primària és el "nick" pel que no hi poden haver dos usuaris amb el mateix "nick". El correu electrònic i el telèfon són de tipus únic el que vol dir que no es poden repetir entre usuaris diferents. Per aquestes raons sí que hi poden haver usuaris amb el mateix nom, cognom, contrasenya, sexe, activació o estatus iguals. El telèfon és un *varchar* en comptes d'un *integer* perquè ja es limita que només es puguin entrar números a l'aplicació i es guarda com a *varchar* a la base de dades. Tant la contrasenya com l'activació tenen un *varchar* tan gran perquè al xifrar-les augmenten la seva mida i necessiten més espai de memòria per poder-les guardar correctament a la base de dades. L'estatus només pot ser 0 o 1 perquè quan és 0 vol dir que el compte no està activat i quan és 1 vol dir que el compte sí que està activat.

2. Taula d'administradors

És la taula encarregada de guardar els administradors, que tal com s'ha explicat abans, poden accedir a algunes funcionalitats extres de l'aplicació.



id6206731 tpclub admins	
🔑	nick_a : varchar(20)

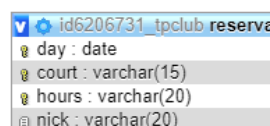
II-lustració 25: Informació de la taula d'administradors

Es pot observar a la imatge de la informació que aquesta taula només té una columna de “nick_a”, que és el “nick” dels usuaris que són administradors i és la clau primària perquè no hi poden haver dos administradors repetits. A més a més, aquesta columna és clau forana de la columna de “nick” de la taula usuari i s'especifica en la creació de la taula.

En la creació de la clau forana s'ha especificat que si es modifica ho es borra la clau a la qual fa referència també es modifiqui o s'elimini en conseqüència, ja que si un usuari que és administrador es canvia el “nick” també ha de canviar a la taula d'administradors, i si s'elimina el compte, també s'ha d'eliminar el seu “nick” a la taula d'administradors.

3. Taula de reserves

Es la taula de la base de dades encarregada de guardar les reserves del club per saber amb exactitud quina reserva ha fet un usuari per un dia, pista i hora concreta.



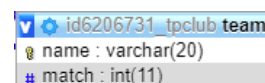
id6206731 tpclub reserva	
🔑	day : date
🔑	court : varchar(15)
🔑	hours : varchar(20)
🔑	nick : varchar(20)

II-lustració 26: Informació de la taula de reserves

Aquesta taula té una clau primària de 3 atributs (dia, pista i hora) de manera que no hi poden haver 2 reserves iguals per la mateixa hora. El dia és de tipus *date*, ja que interessa guardar el dia, mes i any que es reserva una pista. La pista és de tipus *varchar* ja que es una cadena de caràcters per diferenciar les pistes disponibles i la hora és també de tipus *varchar* perquè també és una cadena de caràcters que engloba una sola hora (Ex: 09:00 – 10:00). El “nick” és clau forana de la taula d'usuaris, ja que les reserves només les poden fer les persones registrades dins de l'aplicació. Amb aquesta estructura un usuari pot reservar diferents hores en diferents dies i pistes però mai pot reservar el mateix 2 vegades, ni tampoc reservar una pista ja ocupada per algú altre.

4. Taula d'equips

És una taula utilitzada per emmagatzemar el nom dels equips i el seu número (aquest número serveix per emparellar els equips en partits i per saber si un equip esta actiu o ha perdut).



id6206731 tpclub team	
🔑	name : varchar(20)
#	match : int(11)

II-lustració 27: Informació de la taula d'equips

Dins d'aquesta taula trobem una clau primària d'un sol atribut on es guarda el nom de l'equip, que al ser clau primària, produeix que no es pugui repetir dins d'un torneig però també entre torneigs. Per altra banda el número és una característica de cada equip que si es pot repetir com explicaré més endavant pel control dels torneigs.

5. Taula de pertinences

Taula que apareix de la relació entre la taula d'usuaris i la taula d'equips. Aquesta taula relaciona les claus primàries de cada taula formant la seva clau primària amb la unió d'aquestes 2.

id6206731	tpclub	nick	team
		varchar(20)	varchar(20)

Il·lustració 28: Informació de la taula pertinença

Gràcies a aquesta taula es poden formar equips de més d'una persona, un mateix equip no pot tenir dos jugadors iguals i un jugador sí que pot estar dins d'equips diferents. Per aquesta raó aquest conjunt de taules és extrapolable a poder fer equips de grans dimensions tot i que en aquest cas només es fan equip d'una o dues persones per les característiques dels esports del club.

6. Taula de partits

És una taula on s'emmagatzemen els partits entre equips i que apareix de la relació entre 2 equips tal com es pot veure a l'esquema entitat-relació.

id6206731	tpclub	team1	team2	result	winner	loser	num
		varchar(20)	varchar(20)	varchar(30)	varchar(20)	varchar(20)	int(11)

Il·lustració 29: Informació de la taula de partits

En aquesta taula la clau primària surt de la relació entre 2 equips, és a dir, la clau primària està formada per 2 atributs que són el nom dels equips que juguen el partit. Gràcies a aquesta clau primària no hi poden haver 2 partits iguals entre 2 equips. A més a més, aquesta taula també té 4 atributs informatius de cada partit. El resultat, el guanyador i el vencedor (els 3 *varchar*) inicialment tindran un valor nul fins que es registri la informació del partit. Per altra banda, el número és un *integer* que s'incrementa automàticament quan es creen els partits per tenir l'ordre amb el qual es juguen registrat també. Gràcies a aquesta taula es pot tenir un control en tot moment de com avancen els partits i de les seves modificacions.

7. Taula de torneigs

Taula encarregada de guardar la informació dels torneigs creats per un administrador prèviament.

id6206731	tpclub	name	active	players	type	level	pi	control	sets
		varchar(20)	varchar(5)	varchar(10)	varchar(10)	varchar(20)	varchar(10)	int(11)	varchar(20)

Il·lustració 30: Informació de la taula de torneigs

Aquesta taula té una clau primària formada per un sol atribut que és el nom del torneig utilitzant el tipus *varchar* perquè és una cadena de caràcters. A continuació podem trobar 5 atributs de tipus *varchar* que donen informació sobre com és el torneig. Els 5 atributs són si el torneig està actiu o no, el nombre màxim d'equips (8 equips, 16 equips o 32 equips), el tipus de torneig (mixt, masculí o femení), si és individual o per parelles (en el tennis es pot triar, però en pàdel sempre són per parelles) i el nombre de sets per partit (1 set, 3 sets, 5 sets).

Finalment també hi ha un atribut anomenat control que és de tipus *integer* i que serveix per fer avançar el torneig. Aquest atribut s'explica amb més detall en el disseny dels torneigs.

8. Taula de registres

Taula que relaciona la taula d'equips i la taula de torneigs, és a dir, permet veure quins equips estan registrats a un torneig concret.

id6206731_tpclub	registre
team	varchar(20)
name_t	varchar(20)

Il·lustració 31: Informació de la taula de registres

La clau primària d'aquesta taula està formada per les claus primàries de la taula de torneig i la taula d'equips que al mateix temps són clau forana de la seva taula. Els dos atributs de la clau primària són de tipus *varchar* perquè si guarda el nom de l'equip i el nom del torneig on s'ha registrat.

Tal com està configurada la taula, un torneig pot tenir diferents equips i un equip pot participar en diferents torneigs. Tot i que l'afirmació anterior és certa, de la manera que està fet el registre als torneigs un mateix equip no pot participar en més d'un torneig al mateix temps, tot i això, els jugadors d'un equip sí que poden crear un altre equip amb nom diferent i participar en un altre torneig al mateix temps si en són capaços.

9. Taula de l'historial de torneigs

Aquesta taula és un registre dels torneigs que ja s'han fet on s'especifica la seva durada, les seves característiques i l'equip guanyador un cop finalitzat cada un perquè en quedi constància. No està relacionada amb la taula de torneigs, ja que quan s'acaba un torneig s'elimina de la seva taula i es guarda la informació en aquesta taula.

id6206731_tpclub	h_tournaments
name_t	varchar(20)
data_i	datetime
data_f	datetime
inf	text
winner	varchar(20)

Il·lustració 32: Informació de l'historial de torneigs

La clau primària està conformada per 2 atributs, el primer és el nom del torneig de tipus *varchar* i el segon és la data d'inici de torneig de tipus *datetime*, ja que interessa guardar a part del dia també les hores, minuts i segons per si és un torneig exprés d'un dia. Amb aquesta clau primària no hi podran haver dos torneigs del mateix tipus creats a la mateixa hora. Per altra banda, hi podran haver tants torneigs de tennis i pàdel creats en diferents moments com vulguis.

A part de la clau primària també hi ha 3 atributs ben diferenciats sobre la informació del torneig. El primer és la data de fi del torneig que es manté amb valor nul fins que el torneig finalitza i es registra en format *datetime*, igual que la data d'inici per veure l'interval de temps amb més exactitud. La informació del torneig es registra just quan es crea el torneig i és de tipus "text" perquè és una cadena de caràcters variable i bastant llarga amb tota la informació del torneig.

Finalment l'últim atribut és el guanyador del torneig, que és de tipus *varchar* i es manté a nul fins que finalitza el torneig amb un vencedor. Si el torneig s'acaba abans d'hora per decisió d'un administrador, el guanyador es queda en nul per indicar que el torneig no es va acabar correctament.

10. Taula de la llista negra

Es la taula on es guarda la informació primordial dels jugadors que han sigut expulsats per alguna raó de l'aplicació directament per un administrador.

id6206731_tpclub	ban
nick	varchar(20)
dni	varchar(20)
email	varchar(30)
inf	text

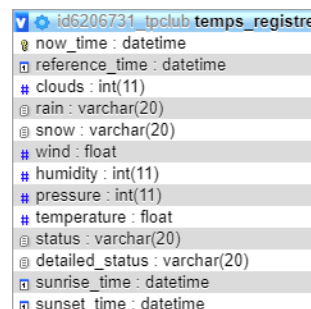
Il·lustració 33: Informació de la taula de la llista negra

La clau primària d'aquesta taula és el DNI de l'usuari expulsat, ja que és un atribut únic de cada ciutadà i no es pot canviar. D'aquesta manera si algú que ha sigut expulsat intenta tornar a registrar-se al posat el seu DNI es consultarà aquesta taula i no es deixarà a aquest usuari registrar-se a l'aplicació. Per complementar una mica la informació de l'expulsat es guarda també el seu "nick" i el seu correu electrònic (utilitzant el tipus *varchar*) per si s'ha de tornar a contactar amb ell més endavant. Encara que es guardi el "nick" si algú altre vol utilitzar el mateix després sí que podrà, ja que no es trobarà a la taula d'usuaris.

Finalment hi ha una columna per guardar la informació de la raó per la qual s'ha expulsat el jugador. Aquest atribut és de tipus text perquè l'explicació pot ser de mida variable i serà una cadena de caràcters que escriurà un administrador.

11. Taula del registre del temps atmosfèric

Taula destinada a guardar tota la informació dels registres que fa el programa de *python* que consulta una llibreria oberta sobre el temps atmosfèric i envia la informació del temps a les pistes cada hora. Aquest programa s'explica en més detall en el disseny del programa de *python* i en la seva implementació.

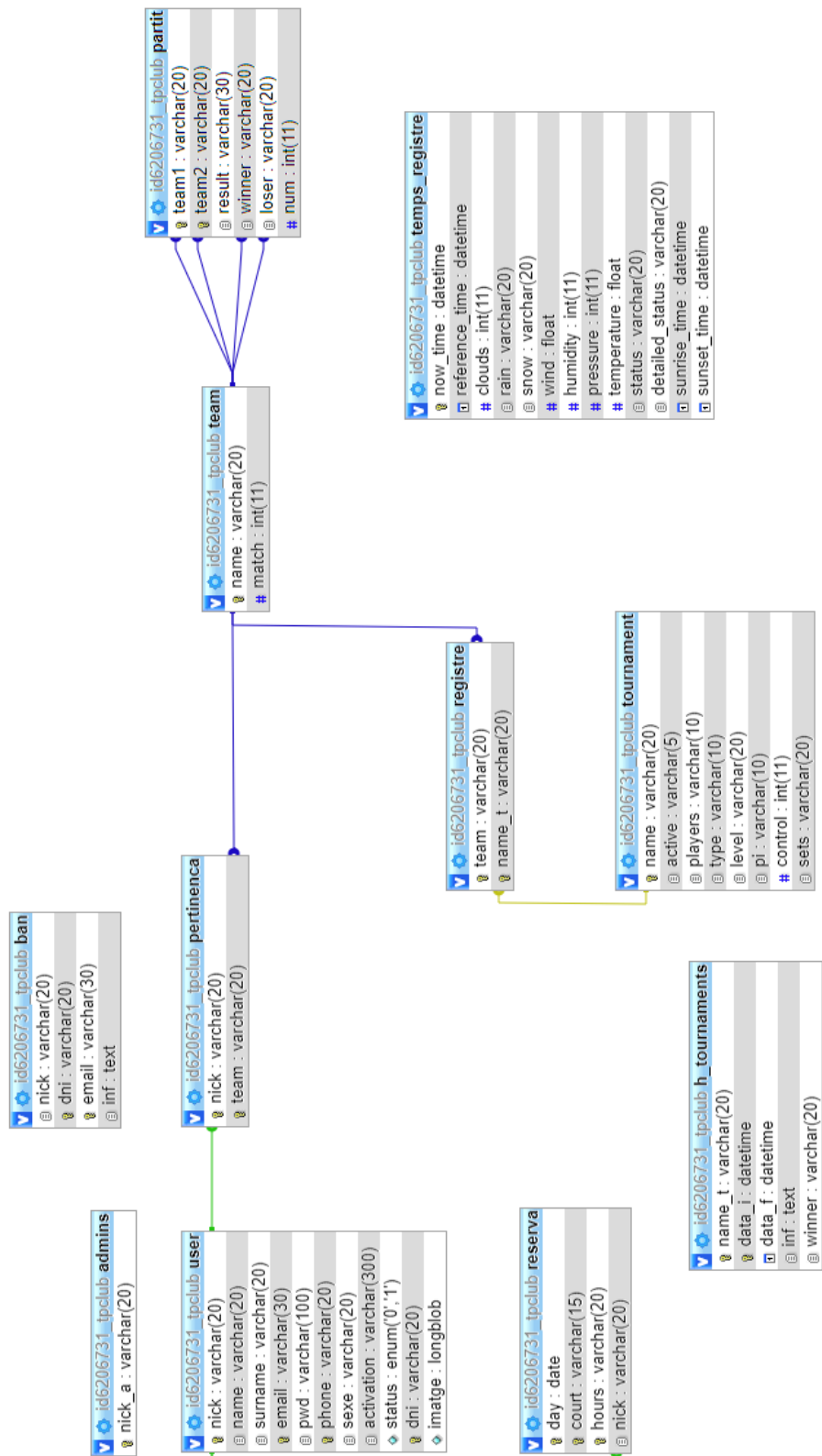


id6206731 tpclub temps_registre
now_time : datetime
reference_time : datetime
clouds : int(11)
rain : varchar(20)
snow : varchar(20)
wind : float
humidity : int(11)
pressure : int(11)
temperature : float
status : varchar(20)
detailed_status : varchar(20)
sunrise_time : datetime
sunset_time : datetime

Il·lustració 34: Informació de la taula del registre atmosfèric

Aquesta taula té molts atributs diferents que donen la informació atmosfèrica per separat. La clau primària és la data actual en format *datetime*. El següent atribut és la data de referència, ja que la llibreria agafa les dades del temps en certs instants i d'aquesta manera es pot veure exactament en quin moment feia el temps enregistrat a la pista. Tots els altres atributs són les diferents característiques atmosfèriques d'interès com els núvols, la pluja, la neu, el vent, la humitat, la pressió, la temperatura, etc. Cada un d'aquests atributs és del tipus que li va més convenient depenent de la informació que guarda.

El *phpMyAdmin* crea un esquema amb les relacions marcades amb fletxes de les taules de la base de dades:



II-lustració 35: Esquema de la base de dades generat per phpMyAdmin

4.7. Disseny de les peticions HTTP i gestió al servidor

En aquest projecte només s'han utilitzat dos tipus de peticions:

Peticions GET

Les peticions GET permeten enviar una petició per poder consultar la informació de les taules de la base de dades i reben un JSON amb la informació demanada depenen de les capçaleres.

Peticions POST

Les peticions POST permeten enviar una petició per poder modificar la informació de les taules de la base de dades i reben una resposta per saber si la modificació s'ha completat correctament o no.

Per a dur a terme la gestió de la base de dades del servidor s'utilitzen 2 fitxers PHP principal per tal de rebre les peticions i retornar una resposta a l'usuari. S'han unificat totes les operacions de modificació, eliminació i inserció dins del fitxer "Modifica.php" on es gestionen totes les peticions de tipus POST. Per altra banda, totes les operacions per obtenir dades es duen a terme amb el fitxer "Consulta.php". Hi ha algun fitxer PHP separat, per fer alguna funció més específica. Tots els fitxers PHP que reben peticions retornen la informació necessària en format JSON.

Fitxers PHP principals

a) Consulta.php

Aquest fitxer és l'encarregat de rebre totes les peticions GET i buscar quina de totes les que engloba s'està demana per poder gestionar-la consultant la base de dades. Per poder diferenciar entre les diferents peticions a dins del fitxer PHP es comprova amb condicionals les capçaleres de les peticions que s'envien utilitzant la funció *isset* com es pot veure en la següent imatge.

```
//Consulta admins
if(isset($_GET["admins"])) ){
    $select="select * from admins";
    $resultat_select=mysqli_query($conexio,$select);
    $i=0;
    while($registre=mysqli_fetch_array($resultat_select)){
        $src=utf8_converter($registre);
        $json['admins'][]=$src;
        $i++;
    }
    if ($i==0){
        $arr=array('nick_a'=>"Cap_consulta");
        $json['admins'][]=$arr;
    }
}

//Consulta usuarios
elseif(isset($_GET["users"])) ){
    $select="select * from user where nick not in (select
        nick_a from admins)";
    $resultat_select=mysqli_query($conexio,$select);
    <i>i=a</i>
```

Il·lustració 36: Codi del fitxer Consulta.php

En aquesta imatge es pot veure clarament la diferenciació de peticions depenent de la capçalera de les peticions, ja que en aquest cas si arriba "admins" retorna un JSON amb la informació de tots els administradors i si arriba "users" retorna un JSON amb la informació de tots els usuaris normals.

b) Modifica.php

Aquest fitxer és l'encarregat de rebre totes les peticions POST i buscar quina de totes les que engloba s'està demana per poder gestionar-la modificant la base de dades. Per poder diferenciar entre les diferents peticions a dins del fitxer PHP es torna a comprovar

amb condicionals les capçaleres de les peticions que s'envien utilitzant la funció *isset* com es pot veure en la següent imatge.

```
//Elimina un equip
elseif(isset($_POST["nameTeamD"])) ){
    $nameTeam=mysqli_real_escape_string($conexio,$_POST['nameTeamD']);
    $delete="delete from team where (name)='{ $nameTeam}'";
    $resultat_delete=mysqli_query($conexio,$delete);
    $affected_rows=mysqli_affected_rows($conexio);
    if($affected_rows>0){
        $resposta="ok";
    }
    else{
        $resposta="no_ok";
    }
}

//Elimina el torneig
elseif(isset($_POST["nameET"])) ){
    $name=mysqli_real_escape_string($conexio,$_POST['nameET']);
    $delete="delete from tournament where (name)='{ $name}'";
    $resultat_delete=mysqli_query($conexio,$delete);
    $affected_rows=mysqli_affected_rows($conexio);
    if($affected_rows>0){
        $resposta="ok";
    }
}
```

Il·lustració 37: Codi del fitxer Modifica.php

En aquesta imatge es pot veure clarament la diferenciació depenent de la capçalera de les peticions, ja que en aquest cas si arriba "nameTeamD" s'elimina l'equip amb el nom que ha arribat. A continuació retorna una resposta per confirmar si s'ha eliminat o no correctament l'equip. Per altra banda si arriba "nameET" s'elimina el torneig que hagi arribat i també es retorna una resposta per confirmar si s'ha eliminat correctament.

En aquest fitxer també hi ha gestions de modificació i d'inserció que funcionen exactament igual que les d'eliminació i es distingeixen per la capçalera com les altres.

Fitxers PHP per propòsits específics

Els fitxers PHP secundaris duen a terme operacions específiques que fan que sigui més còmode que estiguin separats dels fitxers principals.

Nom fitxer	Origen de la petició	Descripció
<u>Send Mail.php</u>	Altres fitxers PHP del servidor.	Funció per enviar correus electrònics als usuaris i que utilitza la llibreria "PHPMailer".
<u>activation.php</u>	Correu electrònic d'un usuari.	Gestiona la petició per activar el compte d'un usuari quan s'envia des del seu correu electrònic al clicar l'enllaç d'activació.
<u>canvi_contra.php</u>	Correu electrònic d'un usuari.	Gestiona la petició per recuperar la contrasenya d'un usuari des del seu correu electrònic en clicar l'enllaç de recuperació.
<u>canvi_info_usuari.php</u>	Correu electrònic d'un usuari.	Gestiona la petició per modificar la informació d'un usuari des del seu correu electrònic al clicar l'enllaç de modificació.

<u>temps.php</u>	Programa de <i>pyhton</i> de gestió meteorològica.	Gestiona les peticions que envia el programa de <i>pyhton</i> i guarda la informació meteorològica que rep a la base de dades.
----------------------------------	--	--

Taula 9: Fitxers PHP per propòsits específics

4.8. Disseny de “scripts” i procediments importants

4.8.1. Disseny de les reserves

El disseny de la reserva s'ha pensat de manera que sigui intuïtiva i simple de dur a terme per un usuari. Primer de tot l'usuari tria un dia dels disponibles en un calendari gràfic, a continuació tria una pista i finalment les hores. Si es vol veure amb detall les pantalles gràfiques i com interactuen està explicat al [manual d'usuari](#).

Un cop s'han registrat les dades claus d'una reserva (data, pista i hores) s'envia una petició POST cap al servidor juntament amb el sobrenom de l'usuari per saber qui ha fet la reserva. Si s'han triat diverses hores, s'envien juntes amb una cadena de caràcters separades per un signe concret, en aquest cas un “_”. D'aquesta manera encara que s'hagin seleccionat més d'una hora s'envien amb una sola petició.

Com que a la base de dades la clau primària és la combinació de dia, pista i hora, en arribar la petició el primer que es fa és separar la cadena de caràcters en les diferents hores i guardar-les per separat a la base de dades. Tot i ser només una petició POST pot retornar diferents respostes depenen de les excepcions i la situació de la reserva que es comprova en el servidor. Hi poden haver 4 respostes possibles a una reserva normal:

- 1) La reserva s'ha completat correctament sense cap problema perquè s'han pogut inserir totes les hores seleccionades correctament a la base de dades.
- 2) La reserva no ha sigut possible perquè alguna o totes les hores seleccionades no s'han pogut inserir correctament a la base de dades.
- 3) Que es detecti a partir de la data seleccionada que és una data molt llunyana (més de 3 mesos) i s'envii una resposta personalitzada per avisar-ho. La possibilitat que es pugui clicar un dia anterior a l'actual ja es controla directament en l'aplicació.
- 4) Que es detecti que s'han intentat inserir més de 4 hores en una mateixa pista un mateix dia, ja que està limitat a com a màxim 4.

Gràcies a aquest disseny i al disseny de la base de dades un usuari en una data i hores concretes només pot reservar en una pista, ja que es té en compte a l'hora de guardar la informació i a més a més, en mostrar-la, les hores ja reservades en una pista (un mateix dia per un usuari) apareixen ocupades a les altres pistes.

Finalment, dir que en una pista, en una data i hora concreta la reserva la fa només un usuari, que serà el responsable de la pista aquella hora. D'aquesta manera un usuari del club pot jugar amb algú que no ho és, es poden fer reserves per jugar per parelles o directament per entrenaments on la reserva la fa un dels integrants (que serà el responsable).

4.8.2. Disseny dels torneigs

En aquest apartat s'explica el disseny dels torneigs, les fases que pot tenir i les diferents singularitats.

Primer de tot s'ha de tenir en compte que un torneig té 2 fases principals. La primera és quan es crea per un administrador i ja es pot començar a omplir pels participants. Un cop s'ha creat aquell torneig està creat però no està actiu, ja que només si poden registrar els participants però encara no han començat els partits. La segona fase es produeix quan el torneig ja està ple i un administrador l'activa provocant així que es creïn automàticament els partits de la primera ronda de manera aleatòria entre els participants. A continuació s'expliquen les diferents fases d'un torneig des que s'inicia fins que es finalitza correctament.

1) Creació d'un torneig

La primera fase és la creació i només la pot dur a terme un administrador. Només es pot tenir un torneig de tennis i un torneig de pàdel creats al mateix temps. Aquesta singularitat s'ha dut a terme per les dimensions del club de tennis, ja que només hi ha 2 pistes de cada tipus i els torneigs ocupen hores que podrien ser reserves normals. Tot i això, de la manera que està feta la base de dades es podrien tenir més de 2 torneigs al mateix temps sense problemes (només canviant el seu nom), però s'ha optat per limitar-ho a 1 de tennis i 1 de pàdel al mateix temps com a màxim.

Un cop un administrador crea un torneig s'envia una petició POST amb tota la informació d'aquest (número de jugadors, tipus de torneig, nivell del torneig...). Per el que s'ha explicat, abans es comprova que no hi hagi un torneig ja creat del mateix tipus i s'hi n'hi ha un no es pot crear un altre fins que es finalitzi l'actual. Al crear-se es posa a la columna d'activació que no està activat, ja que hi ha una columna de la taula de torneigs que indica si el torneig està activat. Al crear-se, també s'omple una columna anomenada control, amb el valor màxim de jugadors del torneig que s'utilitzarà per fer avançar les rondes de partits automàticament i s'explica amb més detall en els següents punts.

2) Inscripcions d'un torneig

Les inscripcions a un torneig canvien una mica si és un torneig individual o per parelles però s'han unificat utilitzant els equips. De manera que sempre hi ha equips en el torneig però poden ser d'una o dues persones. Aquesta funcionalitat s'ha limitat a equips d'1 o 2 persones, ja que en el tennis i el pàdel no hi poden haver equips de més de 2 persones. Tot i que està plantejat a la base de dades de manera que es podrien crear equips de moltes persones i reutilitzar les taules i les relacions entre elles per altres clubs d'esports diferents que necessitessin equips més grans.

Tant en les inscripcions per equips individuals com per parelles es comprova que els usuaris no participin ja en el torneig, que en nom d'equip no existeixi, que el gènere dels usuaris es correspongui amb el tipus de torneig i que el torneig no estigui ple. Si se superen les comprovacions, es crea l'equip a la base de dades i la seva relació amb el torneig que toqui. A cada equip registrat nou se li genera el número 0 a la columna de "match" de la base de dades de moment, ja que només s'està omplint el torneig. Aquest número servirà per fer els emparellaments dels partits.

3) Activació d'un torneig

Un cop on torneig està ple un administrador pot iniciar el torneig per tal d'activar-lo. Al activar el torneig es generen una sèrie de mecanismes de manera seqüencial per tal crear la primera ronda de partits:

- a) Primer de tot es fan comprovacions prèvies per veure si el torneig ja està activat o no està ple encara.
- b) A continuació es generen els números de l'1 al màxim numero de jugadors del torneig i s'insereixen en els diferents equips a la columna de "match" de manera aleatòria.
- c) En aquest pas es consulta el numero de "match" de cada equip del torneig que s'està activant i s'emparellen segons els números consecutius, és a dir, l'1 amb el 2, el 3 amb el 4, etc. Aquesta emparellaments provoquen la creació de la primera ronda de partits entre els diferents equips.
- d) Finalment, quan ja s'han generat tots els partits s'activa el torneig definitivament, modificant la columna d'activació.

Quan ja s'ha activat es poden iniciar els partits entre els participants i aquests poden inserir la informació dels resultats per tal de fer avançar les rondes sense necessitat de la intervenció d'un administrador. Tot i això, si es produeix alguna incidència, l'administrador pot modificar els partits en qualsevol moment.

4) Avançament de les rondes de partits

Per entendre bé aquesta fase s'ha de tenir clara i present la base de dades d'aquest projecte i concretament la taula d'equips, partits i torneig, que s'ha explicat en detall en el [disseny de la base de dades](#).

Cada cop que un usuari modifica un partit actiu on participa, ell pot registrar un resultat i s'hi ha sigut victòria o derrota. D'aquesta manera només es dona el partit per finalitzat quan els dos equips que hi participen confirmen la derrota i la victòria per separat. A més a més, cada cop que algú registra el resultat d'un partit es fan tot un seguit de comprovacions per registrar-lo bé i veure si ja es pot passar a la següent ronda de partits:

- 1) Primer de tot es consulta la informació completa del torneig actual i es guarda en una classe "Torneig" personalitzada per tal d'utilitzar-la en els següents passos.
- 2) Es comprova que el jugador que està intentant fer la modificació participi en el torneig i tingui un partit actiu (és a dir, un partit que el seu equip encara no ha actualitzat). També es comprova que el seu equip estigui "viu", és a dir, que no l'hagin eliminat anteriorment.
- 3) A continuació, si s'ha superat el segon pas, es consulta la informació del partit actual de l'usuari i es guarda en una classe personalitzada "Partit" personalitzada per tal d'utilitzar-la en els següents passos.
- 4) En aquest pas, després de registrar la informació del partit també es comprova si es vol registrar una victòria o una derrota:

Victòria

Es captura la informació entrada del resultat del partit i es comproven les quatre possibilitats en voler registrar una victòria.

- Primer es comprova si ja hi ha registrada la victòria i la derrota comprovant si les columnes de "winner" i "loser" de la taula partit estan ocupades. En aquest cas voldrà dir que el partit ja està registrat i no es continuarà l'execució dels següents passos.
- També es comprova si no hi ha registrat ni el guanyador ni el perdedor. En aquest cas sí que es continuaria amb els següents passos.
- Si ja hi ha registrat el guanyador però no el perdedor vol dir que el guanyador ja s'ha registrat i que no es continuarà l'execució dels següents passos.
- Finalment, si el guanyador no està registrat però el perdedor sí, es comprova que el que vol registrar la victòria no sigui ja el perdedor. En cas contrari, es continua amb els següents passos.

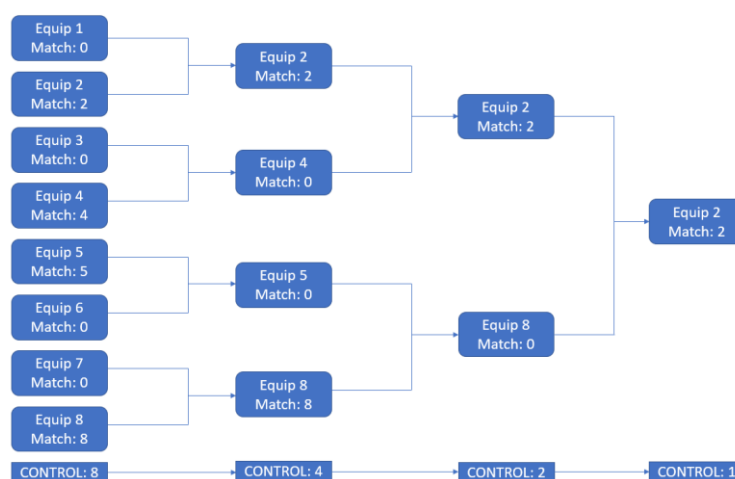
Derrota

Es captura la informació entrada del resultat del partit i es comproven les quatre possibilitats al voler registrar una derrota.

- Primer es comprova si ja hi ha registrada la victòria i la derrota comprovant si les columnes de "winner" i "loser" de la taula partit estan ocupades. En aquest cas voldrà dir que el partit ja està registrat i no es continuarà l'execució dels següents passos.
 - També es comprova si no hi ha registrat ni el guanyador ni el perdedor. En aquest cas sí que es continuaria amb els següents passos.
 - Si ja hi ha registrat el perdedor però no el guanyador vol dir que el perdedor ja s'ha registrat i que no es continuarà l'execució dels següents passos.
 - Finalment, si el perdedor no està registrat però el guanyador sí, es comprova que el que vol registrar la victòria no sigui ja el guanyador. En cas contrari, es continua amb els següents passos.
- 5) Si s'han superat les comprovacions anteriors, es registra l'equip del guanyador o del perdedor del partit actual. En cas de registrar un equip com a perdedor es canvia el seu valor de "match" de la taula equip a 0 per simbolitzar que l'han eliminat. En canvi els

equips guanyadors conserven el seu valor de “match” donat inicialment en generar els primers partits.

- 6) Finalment es comprova el valor de la columna control i es compara amb el nombre de partits finalitzats (entenent partits finalitzats com a partits on ja s’ha registrat tant el guanyador com el perdedor). Utilitzant en nombre de partits finalitzats, es calcula el nombre de partits que falten amb una simple resta i es compara al número de control dividit per 2. Si la comparació dóna igual, es passarà al següent pas on es genera la nova tanda de partits. En cas contrari es finalitza l’execució perquè encara no toca passar de ronda.
- 7) Si es compleix la comprovació anterior, es modifica el valor de control a la seva meitat i es guarda a la base de dades per permetre passar a la següent ronda quan toqui.
- 8) En aquest pas, com que s’ha de generar la nova ronda de partits, es consulten tots els equips no eliminats i s’emparellen un altre cop tenint en compte els números consecutius. D’aquesta manera es pot assegurar que les noves rondes ordenaran bé els partits, tal com es pot veure en el següent esquema. En aquest pas també es comprova si ja hi ha un guanyador del torneig, si es dóna el cas, es registra a l’historial de torneig perquè en quedi constància abans que s’elimini el torneig.



Il·lustració 38: Evolució dels partits en un torneig de 8

Tots aquests passos funcionen exactament igual tant en els torneigs de tennis com en els de pàdel, a més a més s'utilitzen les mateixes funcions, independentment de si és un torneig de 8, 16 o 32 equips. Ja que el valor que fa que hi hagi més rondes de partits (el control) canvia a l'inici del torneig depenen del nombre d'equips màxim.

5) Finalització d'un torneig

El torneig es pot finalitzar en qualsevol moment, tot i que només ho pot fer un administrador. Si es finalitza quan ja s'han fet tots els partits i ha acabat amb normalitat s'ha pogut guardar el guanyador a l'història de torneigs. En canvi, si es finalitza a mig torneig, el guanyador de l'història queda en blanc simbolitzant que el torneig es va quedar a mitges.

Quan es vol finalitzar un torneig se segueixen els següents passos:

- a) Primer de tot es comprova si l'usuari que vol eliminar el torneig és un administrador fent una consulta a la taula d'administradors.
- b) A continuació s'actualitza l'hora i data de finalització a l'història de torneigs.
- c) En aquest pas s'eliminen tots els equips que pertanyen al torneig que s'està eliminant.
- d) Finalment s'elimina el torneig de la seva taula donant la possibilitat de crear-ne un altre del mateix tipus.

4.8.3. Disseny de modificació de partits

Un administrador té la capacitat en tot moment de registrar els partits directament ell o modificar-los un cop ja s'han registrat perquè hi ha hagut algun problema o malentès. En el cas de registrar un partit des de zero, funciona exactament igual que si es registra de manera normal amb els equips de cada partit.

Tot i això, aquesta situació provoca que alguns cops hagi de girar el resultat en el sentit que el guanyador passa a ser el perdedor o al rebés. Quan es provoca una inversió en un partit s'ha de tenir en compte que el número de "match" dels dos equips s'ha de modificar correctament a part de la modificació en el mateix partit a la base de dades. Hi ha diferents situacions que es capturen a l'hora d'invertir un partit per tal de modificar-lo correctament:

- 1) **Si tant el guanyador com el perdedor estan registrats:** en invertir el partit primer de tot es comprova el valor del "match" del guanyador i l'ordre amb el qual estan registrats al partit. D'aquesta manera es pot generar el valor de "match" de l'actual perdedor sumant o restant 1 al valor de "match" consultat depenen de l'ordre dels 2 equips. Un cop obtingut el nou valor de "match" es posa a l'equip perdedor i el guanyador si posa el 0 (simbolitzant que està eliminat). Un cop canviat el valor de "match" correctament en els 2 es fa el canvi en la informació del partit canviant el nom del guanyador pel del perdedor i al rebés.
- 2) **Si només hi ha registrat el guanyador:** en invertir en aquest cas s'ha de posar el "match" d'aquest equip a 0 i només s'ha de canviar el nom de l'equip guanyador i posar-lo en el perdedor.
- 3) **Si només hi ha registrat el perdedor:** en aquest cas en invertir el partit, s'ha de generar el valor antic de "match" buscant l'equip contra el qual juga (encara que no s'hagi registrat com a guanyador encara) i fent el mateix que s'ha explicat en la situació número 1 amb l'única diferència que el "match" de l'equip que encara no s'ha registrat al partit no es modifica.

A més a més, com a administrador, en modificar un partit també es pot buidar el resultat, el guanyador i el perdedor per deixar-lo com si no s'hagués registrat. Si passa això s'han de tenir en compte 2 possibles situacions:

- 1) **Si es vol eliminar el guanyador:** en aquest cas no hi ha molt problema ja que només s'ha de buidar la columna de guanyador del partit sense modificar el "match" de l'equip en qüestió.
- 2) **Si es vol eliminar el perdedor:** aquest cas és una mica més complex, ja que a part de buidar la columna del perdedor del partit també s'ha de restaurar el valor del "match" de l'equip (com que estava registrat com a perdedor se li havia canviat el "match" a 0). La manera de restaurar el seu valor funciona exactament igual que a l'hora d'invertir un partit.

En qualsevol cas s'ha tingut en compte els canvis en cascada, és a dir, si es modifica un partit de la 1a ronda i ja passen per la 3a ronda el guanyador (que és qui ha anat avançant) es modifica en cada una de les rondes de davant fins a arribar a l'actual. Això s'aconsegueix fent la recuperació del seu valor de "match" i modificant la columna del guanyador en el partit actual i en els següents on aparegui. Aquesta funcionalitat es pot dur a terme gràcies al fet que els partits estan ordenats per rondes en la base de dades.

Per tal de modificar el resultat no hi ha problema, ja que es modifica de manera independent si no s'ha modificat ni el guanyador ni el perdedor. Per altra banda, si s'han modificat l'equip guanyador o el perdedor es modifica també el resultat juntament amb el canvi fet utilitzant només una petició.

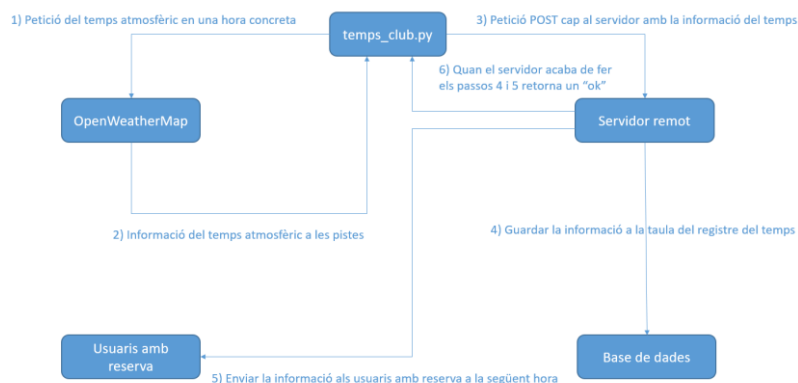
4.8.4. Disseny del programa de “pyhton” per registrar el temps atmosfèric

Aquest programa s’ha dissenyat pensant amb la necessitat que té un usuari de saber el temps que fa a les pistes quan es té una reserva, ja que les pistes del club són exteriors i no s’hi pot jugar amb pluja, neu o altres fenòmens meteorològics que dificultin el correcte funcionament de les pistes.

Aquest programa s’ha dut a terme amb 3 llibreries diferents:

- a) **Llibreria “pyown”**: que permet veure els registres del temps a partir d’unes coordenades o una població. En aquest cas s’utilitzen les coordenades del club de tennis i pàdel de Navàs.
- b) **Llibreria “requests”**: és una llibreria que permet fer peticions HTTP i en aquest cas s’utilitza per fer peticions post cap al servidor remot i aquest s’encarrega de registrar la informació i d’avisar als jugadors que tenen reserva la següent hora.
- c) **Llibreria “time”**: com que la llibreria del temps té limitacions d’accés s’ha optat per fer que es registri el temps a cada hora, concretament al minut 30 de cada hora, ja que d’aquesta manera es pot avisar als usuaris amb 30 minuts d’antelació sobre el temps atmosfèric a les pistes. A més a més, en demanar la informació atmosfèrica de manera tan separada és impossible que hi hagi algun problema pel que fa a les limitacions d’accés a la informació meteorològica.

Utilitzant aquestes 3 llibreries el funcionament és relativament simple. El programa es queda en un bucle infinit i es comprova constantment que no sigui hora de fer una petició. Quan és hora de fer una petició primer es registra el temps actual en aquell moment i just després s’envia la informació cap al servidor remot on es gestionarà. En el servidor remot el primer que es farà en rebre la informació, és registrar-la a la taula de temps atmosfèric de la base de dades i a continuació comprovar quins usuaris tenen una reserva a la següent hora en punt. Un cop trobats els usuaris que tenen una reserva se’ls hi envia un correu electrònic perquè puguin veure el temps atmosfèric que està fent en aquell moment a les pistes.



Il·lustració 39: Diagrama de comunicacions del programa de “pyhton” i el servidor

Finalment un cop el servidor ha complert l’acció de registrar la informació a la base de dades i enviar els correus electrònics als usuaris amb reserves a la següent hora acaba enviant un “ok” al servidor per donar per acabades les accions per aquella hora. Aquest cicle es repetirà a cada hora al minut 3 durant les hores que estigui obert el club de tennis.

4.9. Disseny de la seguretat

4.9.1. Disseny de la seguretat del client

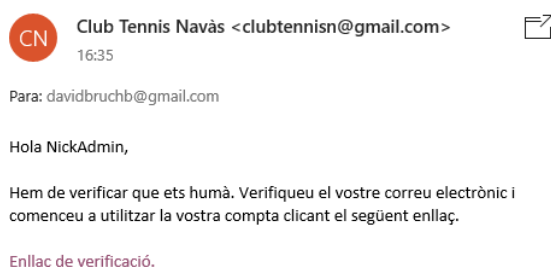
Per aconseguir una seguretat alta i robusta del compte d'usuari i de les dades del client s'han dut a terme diferents accions com la necessitat d'activar el compte a través del correu electrònic un cop s'ha fet un registre, la necessitat de confirmar qualsevol canvi en la informació d'un usuari a través del correu electrònic i el xifratge de la contrasenya de manera que es impossible saber la contrasenya ni tan sols mirant directament a la base de dades.

1. Activació del compte després del registre

L'activació dels comptes serveix per evitar que alguna persona pugui crear comptes en nom d'altres persones per suplantar identitats i també evita la creació massiva de bots, ja que sense necessitat d'activació es podrien registrar massivament comptes falses i això podria provocar greus problemes al correcte funcionament de l'aplicació i del servidor. En canvi, gràcies a l'activació per correu electrònic (que és el sistema que fan servir la gran majoria de pàgines web i aplicacions) és indispensable que cada usuari activi el seu compte d'usuari manualment a través del seu propi correu electrònic, al qual només hi pot accedir ell.

A la taula d'usuaris de la base de dades hi ha dos atributs encarregats exclusivament de l'activació del compte un cop s'ha registrat un usuari, que són l'atribut d'activació i l'atribut d'estatus. Inicialment quan un usuari es registra a través de l'aplicació al finalitzar el formulari de registre i acceptar-lo es crea la seva entrada a la base de dades però amb l'estatus a 0 i amb un valor d'activació que es crea xifrant el correu electrònic de l'usuari més una marca de temps utilitzant "md5" per xifrar-ho. Un cop es té el valor d'activació xifrat s'envia cap al correu de l'usuari amb un enllaç que pot clicar per activar el seu compte.

Email de verificació



Il·lustració 40: Imatge de el correu electrònic d'activació del compte

A continuació, si l'usuari clica l'enllaç d'activació, s'envia una petició cap al servidor remot que comprova si el valor xifrat d'activació, que l'usuari ha retornat des del correu, coincideix amb el valor guardat prèviament a la base de dades. Si coincideix, s'elimina de la base de dades el valor d'activació i es canvia el valor de l'estatus de 0 a 1 indicant que el compte de l'usuari ja està activat i apareix un missatge confirmant que s'ha activat el compte correctament.

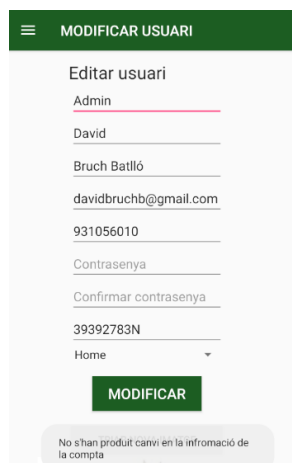
Per altra banda si en comparar els valor d'activació no coincideixen, apareix un missatge avisant del fet que el codi d'activació és erroni. A més a més, també es té en compte si un usuari vol intentar clicar repetides vegades l'enllaç d'activació del correu perquè si es torna a clicar després d'haver activat el compte, es detecta i apareix un missatge especificant-ho a l'usuari. En qualsevol dels dos casos anteriors no s'arriba a modificar la base de dades perquè es detecta abans que el codi d'activació és erroni o el compte ja està activat consultant el registre de l'usuari.

2. Confirmació de qualsevol canvi en la informació del compte

Aquesta confirmació serveix perquè un usuari tingui més seguretat a l'hora de canviar la informació del compte de manera que quan vulgui fer un canvi, per dur-lo a terme, hagi d'entrar dins del seu correu personal per acceptar-lo. Aquest sistema té la seguretat que s'ha d'entrar al

correu electrònic personal, ja que dins d'aquest només hi té accés l'usuari i per això podem assegurar que el que confirma els canvis és ell mateix.

Primer de tot un usuari pot entrar dins de la pantalla d'editar usuari per modificar-se la informació que vulgui de l'usuari actual. En entrar-hi, l'usuari es troba la informació del seu compte, carregada i a la vista menys la seva contrasenya (per seguretat) de manera que pot veure com té la informació ara i canviar-la si vol. Si no canvia la informació i clica editar usuari apareix un missatge advertint que no s'ha produït cap canvi i no s'envia cap petició.



The screenshot shows a mobile application interface with a green header bar containing a hamburger menu icon and the text 'MODIFICAR USUARI'. Below the header, the title 'Editar usuari' is displayed. The form contains several input fields with the following values: 'Admin' (highlighted with a red underline), 'David', 'Bruch Batlló', 'davidbruchb@gmail.com', '931056010', 'Contrasenya', 'Confirmar contrasenya', and '39392783N'. There is a dropdown menu labeled 'Home' with a downward arrow. A green button labeled 'MODIFICAR' is positioned below the form. At the bottom, a light gray message box states: 'No s'han produït canvi en la informació de la compta'.

Il·lustració 41: Missatge advertint que no s'ha produït cap canvi

Per altra banda si s'ha produït algun canvi i es clica el botó de modificar s'envia una petició cap al servidor amb la nova informació de l'usuari i apareix un missatge confirmant que s'ha enviat un correu de confirmació al correu antic. A més a més, just quan es produeix l'acció d'enviar la petició amb els canvis i apareix el missatge, l'aplicació tanca la sessió de l'usuari i el porta cap al menú principal.



The screenshot shows a mobile application interface with a green header bar containing a hamburger menu icon and the text 'MENÚ PRINCIPAL'. Below the header, there are six icons arranged in a 3x2 grid, each with a corresponding green button: 'INICI SESSIÓ' (key icon), 'RESERVAR' (calendar icon), 'TORNEIG T.' (tennis racket icon), 'TORNEIG P.' (paddle icon), 'UBICACIÓ' (location pin icon), and 'TEMPS' (cloud with rain icon). At the bottom, a light gray message box states: 'S'ha enviat un correu de confirmació al correu antic per confirmar el canvis'.

Il·lustració 42: Missatge confirmant que s'ha enviat el correu

Un cop el servidor rep la informació, aquest envia un correu al correu electrònic antic (en cas que se l'hagi canviat) perquè confirmi els canvis fets. L'usuari rep un correu electrònic que diu que s'han produït canvis en el seu compte, de manera que si és veritat només ha de clicar l'enllaç per confirmar-los i si és mentida ha d'ignorar el correu i si pot canviar la contrasenya.

Confirmació dels canvis fets en la compta del club de tennis i pàdel



Il·lustració 43: Correu electrònic de confirmació del canvis

D'igual manera que amb l'activació també s'ha tingut en compte que l'usuari vulgui clicar repetides vegades l'enllaç de confirmació de manera que els canvis només es produiran el primer cop i tots els altres apareix un missatge avisant a l'usuari que els canvis ja s'han confirmat anteriorment.

3. Xifratge de la contrasenya

També s'ha tingut en compte el xifratge de la contrasenya, ja que és una part fonamental de la seguretat del compte d'un usuari. A més a més, no s'ha xifrat amb "md5" ja que se li ha detectat alguna vulnerabilitat i ja no és del tot segura. Per aquesta raó s'ha optat per utilitzar un "password hash" en mode "PASSWORD_BCRYPT" que és més segur actualment.

```
$pwd_enc=password_hash($pwd,PASSWORD_BCRYPT); // encrypted password
```

Il·lustració 44: Xifratge de la contrasenya

Utilitzar l'algoritme "BCRYPT" per crear el "hash" produirà un "hash" estàndard compatible amb "crypt()" i utilitzant un identificador "\$2y\$". El resultat sempre serà una cadena de caràcters de 60 caràcters o "FALSE" en cas d'error.

Per aquesta raó sempre que un usuari es registra o canvia la seva contrasenya editant la seva informació es xifrarà la contrasenya just abans de guardar-la a la base de dades, de manera que si algú pot consultar la base de dades, pot veure la informació dels usuaris però no pot aconseguir la contrasenya dels seus comptes, pel que els seus comptes continuen sent segurs.

A més a més, quan s'hagi de validar la contrasenya perquè un usuari està iniciant sessió tampoc es pot veure la contrasenya sense xifrar, ja que es compara directament xifrada amb la contrasenya introduïda utilitzant la següent funció.

```
password_verify($pwd,$pwd_enciptada_db)
```

Il·lustració 45: Verificació de la contrasenya

D'aquesta manera un cop xifrada la contrasenya a dins de la base de dades no es pot tornar a veure el seu valor directament i només l'usuari que l'ha creat coneix la seva contrasenya.

4. Recuperació de la contrasenya

La recuperació de la contrasenya és una part molt important, per mantenir la seguretat en un compte i fer que un usuari no la pugui perdre del tot si s'ha oblidat de la seva contrasenya. Per la raó que s'ha explicat abans, que la contrasenya un cop es guarda a la base de dades es consulta sempre xifrada, la recuperació és molt important, ja que ni consultant directament la base de dades es pot saber la contrasenya d'un usuari.

La recuperació de la contrasenya no et recupera la teva contrasenya antiga, sinó que en crea una de nova perquè es pugui confirmar des del correu electrònic i així poder tornar a entrar en el compte per canviar la contrasenya a la qual l'usuari vulgui.

Els passos a seguir són molt simples, primer de tot a la pantalla d'inici de sessió hi ha una opció per si t'has oblidat de la contrasenya. Quan l'usuari cliqui aquesta opció hi ha d'haver posat el "nick", ja que és el que es farà servir per guardar la nova contrasenya correctament. Si es clica l'opció de la recuperació de la contrasenya del compte i no hi ha cap "nick" apareix un missatge especificant-ho perquè l'usuari ho vegi.

The screenshot shows a mobile app interface for login. At the top is a green header with a hamburger menu icon and the text 'INICI DE SESSIÓ'. Below the header, the title 'Inici de sessió' is centered. There are two input fields: 'Nick' and 'Contrasenya'. Below the 'Contrasenya' field is a link 'T'has oblidat de la contrasenya?'. Underneath this link are two green buttons: 'TANCAR SESSIÓ' and 'INICI SESSIÓ'. Below these buttons are two links: 'REGISTRAR USUARI' and 'DONAR DE BAIXA'. At the bottom, there is a light gray rounded rectangle containing the text: 'Introdueix el nick per recuperar la contrasenya'.

Il·lustració 46: Missatge avisant que el "nick" esta buit

Si s'introdueix un "nick" primer apareix un quadre de diàleg que demana confirmació per enviar el correu electrònic a l'usuari el qual pertany el "nick" introduït. Si es cancel·la, no passa res però si s'accepta i el "nick" no existeix dins del club apareix un missatge avisant-ho tal com es pot veure en les següents imatges.

The screenshot shows a confirmation dialog box with a white background and a gray border. The title is 'Confirmació'. The text inside says: 'Confirmes que vols rebre i canviar la contrasenya per la contrasenya de recuperació?'. At the bottom right, there are two buttons: 'CANCEL·LAR' and 'CONFIRMAR'.

Il·lustració 47: Diàleg de confirmació per la contrasenya de recuperació

The screenshot shows the same login screen as before. The 'Nick' field now contains the text 'user'. Below the input fields, the same buttons and links are visible. At the bottom, a light gray rounded rectangle contains the text: 'Aquest nick no existeix dins d'aquest club. Repasa el nick.'

Il·lustració 48: Missatge d'avís de que no existeix un usuari amb aquest "nick"

Un cop introduït un “nick” d’un usuari existent i confirmant el quadre de diàleg s’envia una petició cap servidor remot que la gestiona creant una contrasenya aleatòria de 10 dígit i enviant la nova contrasenya al correu electrònic de l’usuari. Un cop l’usuari rep el correu pot llegir directament la nova contrasenya en ell i pot provocar que passi a ser aquella la contrasenya clicant a l’enllaç. D’aquesta manera *només* amb la confirmació a través del correu es canvia la contrasenya, si no es manté l’anterior. Finalment, en clicar l’enllaç s’envia la nova contrasenya al servidor remot, es xifra de la mateixa manera que s’ha explicat abans i es guarda a la base de dades ja xifrada.



Il·lustració 49: Missatge de que s’ha enviat el correu correctament

5. Possibilitat de varies sessions d’un mateix usuari obertes al mateix moment

Aquesta possibilitat s’ha dut a terme per la comoditat dels usuaris, ja que si algun cop volen fer una reserva des d’un altre mòbil amb el seu compte, ho puguin fer. A més a més, qualsevol canvi que es faci s’actualitzarà a les altres sessions quan es carregi qualsevol tipus d’informació, ja que totes miren d’una sola base de dades pel que no comporta un problema de seguretat.

6. Possibilitat de detecció de robatori de compte i recuperació d’aquesta

De la manera que s’ha fet la seguretat del canvi de dades d’un usuari si un usuari malintencionat aconseguís el “nick” i la contrasenya d’un altre i intenta canviar el correu electrònic o qualsevol altra dada important li arribarà l’avís del canvi de dades a l’usuari real del compte al seu correu electrònic i si no les confirma, no es duren a terme. De la mateixa manera, si l’usuari malintencionat comença a reservar pistes només per molestar amb el compte robat, també arribaran correus a l’usuari real.

Tenint en compte això, si un usuari veu moviments que ell no ha fet a través del seu correu electrònic i com que per canviar qualsevol dada s’ha de confirmar pel correu original de l’usuari real, aquest sempre pot entrar igualment al seu compte (ja que es permet l’inici de sessió simultània en diferents mòbils) i canviar la contrasenya confirmant-ho aquest cop per expulsar l’usuari malintencionat del compte infectat.

Finalment remarcar un cop canviada la contrasenya s’expulsarà automàticament a l’usuari malintencionat tancant-li la sessió si intenta fer qualsevol moviment de canvi de pantalles o peticions.

4.9.2. Disseny de la seguretat del servidor i la base de dades

1. Autenticació del servidor web necessària

Per poder accedir al servidor web, utilitzant qualsevol tipus de peticions, les peticions s’hauran d’autenticar amb un nom i una contrasenya prèviament introduïts al servidor de *000webhost* per evitar que qualsevol persona pugui executar els fitxers PHP del servidor. Els passos per protegir el servidor són relativament simples, ja que només s’ha d’entrar a les configuracions de *000webhost* i activar la seguretat per contrasenya dels fitxers PHP. En aquest cas la protecció s’ha dut a terme en el directori “/bd_remota/”, ja que és on hi ha els fitxers PHP executables a partir de les peticions.

Manage password protection

Password protection allows you to require a username and password to access a folder within your site from the web. When password protecting a folder, there are a few things to remember. Protecting a folder will protect all folders within that folder. Also, you will need to create users who can access the protected directory. If folder does not exist it will be automatically created.

Protected directory	Username	
/bd_remota/	xemnas	DELETE

Add

II·lustració 50: Protecció dels fitxer PHP amb contrasenya activada

Per altra banda, des de l'aplicació *Android*, cada una de les peticions que s'enviïn al directori protegit del servidor web hauran de tenir l'autenticació actualitzada amb el nom d'usuari i la contrasenya correctes perquè s'hi pugui accedir.

```
@Override
public Map<String, String> getHeaders() {
    Map<String, String> headers = new HashMap<>();
    String usuari=getString(R.string.user);
    String contrasenya=getString(R.string.pwd);
    String credentials=usuari+":"+contrasenya;
    String auth = "Basic "
        + Base64.encodeToString(credentials.getBytes(), Base64.NO_WRAP);
    headers.put("Authorization", auth);
    return headers;
}
```

II·lustració 51: Part de codi necessària per autenticar les peticions

II·lustració 52: Exemple de que demana autenticació des del navegador

2. Protecció contra injeccions SQL

Es controla en totes les interaccions amb la base de dades que no hi hagi injeccions SQL que pugui realitzar un usuari malintencionat mitjançant la transformació de les sentències amb “`MySQLi_real_escape_string`” explicat anteriorment a l'apartat [d'anàlisi sobre la seguretat per accedir a la base de dades](#).

3. Peticions HTTP o HTTPS

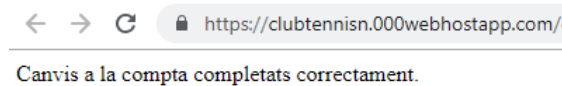
En aquest cas s'han dut a terme peticions HTTPS per enviar les peticions al servidor ja que d'aquesta manera la comunicació està xifrada i porta un certificat SSL d'autenticació que es valida en la resposta de les peticions.

Per poder validar el certificat SSL pots indicar a *HttpsURLConnection* que confii en un conjunt específic de CA (certificats). El procediment pot ser complex. Per això, En el següent exemple s'utilitza CA específica (la del *000webhost* en aquest cas) des d'un *InputStream* i es fa servir per crear un *keystore*, que després s'empra per crear i inicialitzar un *TrustManager*. Un *TrustManager* és el que fa servir el sistema per validar certificats del servidor i, en crear un des d'un *keystore* amb una o més CA; les CA en qüestió seran les úniques en què confiarà aquest *TrustManager*.

Donat el nou *TrustManager*, l'exemple inicialitza un nou *SSLContext* que proporciona un *SSLConnectionFactory* que es pot fer servir per anul·lar el *SSLConnectionFactory* predeterminat de la *HttpsURLConnection*. D'aquesta manera, la connexió farà servir les teves CA per validar certificats.

```
// Validar Certificat
private void validarSSL() throws CertificateException, IOException, KeyStoreException, NoSuchAlgorithmException, KeyManagementException {
    // Load CAs from an InputStream
    CertificateFactory cf = CertificateFactory.getInstance("X.509");
    // From https://clubtennisn.000webhostapp.com/
    InputStream caInput = new BufferedInputStream(new FileInputStream("name: \"000webhostapp.com.crt\""));
    Certificate ca;
    try {
        ca = cf.generateCertificate(caInput);
        System.out.println("ca=" + ((X509Certificate) ca).getSubjectDN());
    } finally {
        caInput.close();
    }
    // Create a KeyStore containing our trusted CAs
    String keyStoreType = KeyStore.getDefaultType();
    KeyStore keyStore = KeyStore.getInstance(keyStoreType);
    keyStore.load(stream: null, password: null);
    keyStore.setCertificateEntry(alias: "ca", ca);
    // Create a TrustManager that trusts the CAs in our KeyStore
    String tmfAlgorithm = TrustManagerFactory.getDefaultAlgorithm();
    TrustManagerFactory tmf = TrustManagerFactory.getInstance(tmfAlgorithm);
    tmf.init(keyStore);
    // Create an SSLContext that uses our TrustManager
    SSLContext context = SSLContext.getInstance("TLS");
    context.init(keym: null, tmf.getTrustManagers(), random: null);
    // Tell the URLConnection to use a SocketFactory from our SSLContext
    URL url = new URL(spec: "https://clubtennisn.000webhostapp.com/bd_remota/");
    HttpURLConnection urlConnection =
        (HttpURLConnection)url.openConnection();
    urlConnection.setSSLSocketFactory(context.getSocketFactory());
}
```

Il·lustració 53: Validació del certificat SSL per la petició HTTPS



Il·lustració 54: Exemple per veure que la connexió és HTTPS

5. Implementació

En aquest apartat de la memòria es mostren les principals parts del projecte i les seves implementacions. Per aquesta raó, es mostraran petites parts del codi per mostrar com s'han implementat les diferents funcionalitats.

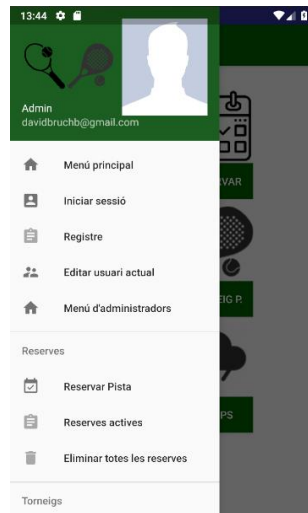
5.1. Part del client

En aquest apartat es tracten les funcionalitats més importants de la implementació del disseny de l'aplicació *Android*.

Primer de tot s'explica la implementació del menú desplegable lateral, l'arxiu manifest, l'arxiu *build.gradle*, el directori *assets*, el directori *res* i l'arxiu *strings*. A més a més, també es mostra com s'han implementat els permisos necessaris per poder fer fotografies i accedir a la galeria, com interactuen els diferents elements de la interfície gràfica, la comunicació amb *Volley*, les tasques asíncrones i el *Shared preferences*.

5.1.1. Menú lateral (*Navegation Drawer*)

Permet a l'usuari desplaçar-se de manera ràpida, senzilla i eficaç a través de totes les pantalles de l'aplicació. És un menú lateral en forma de llista que s'obre des del costat esquerre de l'aplicació ocupant més de la meitat de la pantalla quan es desplega. Es pot desplegar clicant el botó de dalt a l'esquerre disponible a totes les pantalles o arrossegant el dit des de l'esquerra cap a la dreta també a qualsevol pantalla.



II·lustració 55: Menú desplegable lateral

El menú desplegable permet canviar entre les diferents pantalles de l'aplicació de manera ràpida i eficaç només clicant l'apartat al qual el vol obrir.

Per implementar aquest menú primer s'ha d'implementar a l'activitat principal per tal que es pugui veure en totes les pantalles tal com es pot veure en la següent imatge.

```
public class MainActivity extends AppCompatActivity
    implements NavigationView.OnNavigationItemSelectedListener
```

II·lustració 56: Implementació del menú desplegable a l'activitat principal

A continuació quan s'inicia l'aplicació es genera el menú a la funció "onCreate()", que és la primera que s'executa a l'iniciar.

```
Toolbar toolbar = findViewById(R.id.toolbar);
setSupportActionBar(toolbar);

DrawerLayout drawer = findViewById(R.id.drawer_layout);
ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
    activity, drawer, toolbar, "Open navigation drawer", "Close navigation drawer");
drawer.addDrawerListener(toggle);
toggle.syncState();

NavigationView navigationView = findViewById(R.id.nav_view);
navigationView.setNavigationItemSelectedListener(this);
```

II·lustració 57: Generació del menú lateral desplegable

Hi ha una funció anomenada "onNavigationItemSelectedListener()" que escolta en tot moment si es clica algun dels apartats disponibles del menú lateral. Quan es clica la funció captura l'opció que s'ha clicat i actua en conseqüència.

```
@SuppressWarnings("StatementWithEmptyBody")
@Override
public boolean onNavigationItemSelectedListener(@NonNull MenuItem item) {
    int id = item.getItemId();
    displaySelectedScreen(id);
    return true;
}
```

II·lustració 58: Funció que captura l'opció seleccionada

Un cop seleccionada l'opció es genera el seu *id* i es crida una última funció que compara l'*id* amb les opcions disponibles. Quan troba l'opció que encaixa comprova que l'usuari hagi iniciat sessió (si es necessari) i sobreposa el fragment o pantalla que toca.

```
private void displaySelectedScreen(int id){
    Fragment fragment = null;
    if (id == R.id.nav_menu){
        fragment = new MenuP();
    }
    else if (id == R.id.nav_login) {
        fragment = new Login();
    }
    else if (id == R.id.nav_res) {
        if (usuari_actual.equals("")){
            Toast.makeText( context: this, text: ""+ "Inicia sessió o registra usuari abans!", Toast.LENGTH_SHORT).show();
        }
        else {
            fragment = new Calendari();
        }
    }

    if (fragment != null){
        FragmentManager fragmentManager = getSupportFragmentManager();
        FragmentTransaction ft = fragmentManager.beginTransaction();
        ft.replace(R.id.screen_area, fragment).addToBackStack( "tag" );
        ft.commit();
    }
    DrawerLayout drawer = findViewById(R.id.drawer_layout);
    drawer.closeDrawer(GravityCompat.START);
}
```

Il·lustració 59: Parts de la funció que tria el fragment depenen de la id

Per acabar, dins de la funció s'utilitza un condicional per comprovar que el fragment no es nul i s'actualitza. D'aquesta manera no s'ha de crear un nou *FragmentManager* i *FragmentTransaction* dins de cada condicional. Els fragments i activitats de l'aplicació *Android* s'expliquen amb més detall al [Annex B](#).

5.1.2. Arxiu Manifest

L'arxiu *AndroidManifest.xml* és l'arxiu principal de l'aplicació i es troba en el directori arrel. A més a més, és el lloc on declaren els permisos de l'aplicació.

- **Accés a internet:** necessari per a la comunicació entre l'aplicació i el servidor remot.
- **Accés a la càmera:** permet a l'aplicació obrir la càmera i s'utilitza per guardar les imatges dels jugadors.
- **Accés de lectura i escriptura:** permet a l'aplicació consultar o guardar informació (en aquest cas imatges) a l'emmagatzematge del mòbil.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.tfg">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.CAMERA" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />

</manifest>
```

Il·lustració 60: Permisos del arxiu Manifest

Tot i això, a causa de les noves normes de privacitat les opcions que permeten accés a la càmera i a l'emmagatzematge del telèfon no són suficients a partir de la versió M en endavant d'*Android*. En la nova políctica s'especifica que es necessita el consentiment de l'usuari, és a dir, que se li demani amb un quadre de text si està d'acord a permetre que l'aplicació tingui accés a aquestes funcionalitats.

5.1.3. Arxiu build.gradle

En aquest arxiu es troba una molt important del projecte, ja que s'especifiquen paràmetres com per exemple les llibreries, la versió del codi, quina versió d'*Android* és necessària, etc.

```

android {
    compileSdkVersion 28
    defaultConfig {
        applicationId "com.example.tfg"
        minSdkVersion 15
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

```

Il·lustració 61: Codi principal del arxiu build.gradle

La versió mínima del SDK és la 15 i la màxima és la 28 tal com es pot veure en la imatge anterior. La mínima versió es correspon a l'Android 4.0.3 (Ice Cream Sandwich) i la màxima a l'última versió d'Android (Android P).

El paquet d'Android de suport de la compatibilitat disposa de diferents llibreries i dependències que es poden utilitzar dins de l'aplicació tot i que no venen integrades directament. Es poden utilitzar per dotar l'aplicació de funcionalitats noves compatibles amb les versions anteriors, proporcionant diferents funcionalitats i elements que abans no es trobaven disponibles.

```

dependencies {
    api 'com.android.volley:volley:1.1.0'
    api 'pub.devrel:easypermissions:0.2.1'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
    implementation fileTree(dir: 'libs', include: ['*.jar'])
}

```

Il·lustració 62: Dependències utilitzades

5.1.4. Directori assets

És un directori que permet guardar arxius que quan es genera el paquet .apk conserven el nom i les característiques. Per aquesta raó es poden guardar gran quantitat de dades o arxius que no interessa que es modifiquin.

En aquest cas només s'hi guarda la informació del certificat SSL de la pàgina web del servidor per a poder fer la validació de les peticions HTTPS.

5.1.5. Directori res

És el directori encarregat d'emmagatzemar tots els recursos relacionats amb el disseny gràfic i es pot dividir en 5 parts ben diferenciades.

- **drawable:** en aquest directori es troba tota la informació del contingut gràfic que utilitza l'aplicació.
- **layout:** directori on es guarden tots els arxius XML que defineixen el disseny gràfic de les pantalles de l'aplicació.
- **menu:** és un directori que es crea automàticament quan s'importa el menú lateral desplegable i conté el seu document XML i les diferents opcions.
- **mipmap:** conté els diferents logotips que s'utilitzen a l'aplicació.
- **values:** és on es guarden tots els valors dels colors, dimensions i cadenes de caràcters que s'utilitzen a l'aplicació. A l'arxiu "strings.xml" es guarda la informació repetitiva com per exemple la URL del servidor remot ja que d'aquesta manera quan s'hagi de modificar només s'ha de canviar en aquest arxiu.

5.1.6. Permisos de la càmera, lectura i escriptura

Tal com ja s'ha comentat a l'apartat de l'[Arxiu Manifest](#) a causa de les noves normes de privacitat les opcions que permeten accés a la càmera i a l'emmagatzematge del telèfon no són suficients a partir de l'aplicació M en endavant d'*Android*. Per aquesta raó s'ha creat una funció que comprova si en necessiten permisos i si és així els demana quan la versió és igual o superior a la M, si no, s'executa directament, ja que només necessita els permisos del *Manifest*.

```
public boolean validarPermisos(){
    if (Build.VERSION.SDK_INT<Build.VERSION_CODES.M){
        return true;
    }
    if ((checkSelfPermission(CAMERA)== PackageManager.PERMISSION_GRANTED) && (checkSelfPermission(WRITE_EXTERNAL_STORAGE)==PackageManager.PERMISSION_GRANTED)){
        return true;
    }
    if ((shouldShowRequestPermissionRationale(CAMERA))){
        cargar_dialog_permisos();
    }
    else {
        requestPermissions(new String[]{CAMERA,WRITE_EXTERNAL_STORAGE}, requestCode: 100);
    }
    return false;
}
```

Il·lustració 63: Funció que gestiona els permisos de la càmera, de lectura i escriptura

En la funció es pot observar que es comproven diferents aspectes utilitzant condicionals:

- En el primer condicional es comprova si la versió d'*Android* que està utilitzant l'aplicació és inferior a la M. Si és així, amb els permisos del Manifest ja n'hi ha suficient i es continua l'execució normal del programa.
- En el segon condicional es comprova si ja s'han acceptat els permisos anteriorment, ja que si s'accepten un cop, es queda guardada la decisió.
- El tercer condicional comprova si s'han cancel·lat anteriorment els permisos. Si és així, es demana amb un nou diàleg que s'acceptin per poder utilitzar les funcionalitats necessàries.
- L'últim condicional s'executa si encara no hi ha cap decisió guardada d'acceptar o no els permisos. Per aquesta raó demana a l'usuari si els vol acceptar o no.

5.1.7. Interaccions amb els elements de la interfície gràfica

Per interactuar directament amb per exemple un botó, primer s'ha de declarar un objecte Java del tipus *Button* i utilitzant la funció *findViewById* es pot associar la seva representació gràfica del document XML amb l'objecte Java.

```
foto = VG.findViewById(R.id.FOTO);
foto.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        mostrar_dialog_opcions_foto();
    }
});
```

Il·lustració 64: Exemple de l'associació d'un botó

En el cas d'aquest exemple primer s'ha creat l'objecte foto, que és de tipus *Button* i es genera una funció que captura si algú clica el botó i executa el que tingui a dins (en aquest cas una funció per mostrar un diàleg d'opcions).

Una interacció important és la de la validació del DNI quan un usuari l'introdueix per registrar-se al club, ja que ha de tenir la llargada correcta i els números han de quadrar amb la lletra. A més a més, en aquests casos també es comprova que el correu electrònic tingui un format correcte.


```

public boolean validar(String dni){
    String lletraMajuscula="";
    if (dni.length() !=9 || !Character.isLetter(dni.charAt(8))){
        return false;
    }
    lletraMajuscula = (dni.substring(8).toUpperCase());
    return nomesNumeros(dni) && lletraDNI(dni).equals(lletraMajuscula);
}

```

Il·lustració 65: Funció per comprovar que el DNI és correcte

```

public boolean emailPattern(String mail){
    Pattern patternEmail = Patterns.EMAIL_ADDRESS;
    return patternEmail.matcher(mail).matches();
}

```

Il·lustració 66: Funció per comprovar que el correu electrònic és correcte

5.1.8. Comunicació amb el servidor utilitzant Volley

La llibreria de *Google* que permet enviar peticions HTTPS des de l'aplicació cap a un servidor remot i escoltar les respostes depenen de la petició és la llibreria de *Volley*. Perquè pugui funcionar correctament és imprescindible afegir els permisos necessaris de connexió a internet dins de l'arxiu Manifest. Les dues peticions utilitzades són la GET per poder consultar dades de la base de dades i la POST per modificar dades de la base de dades.

Petició GET

Primer de tot s'inicialitza la petició GET especificant l'URL del servidor remot i la petició en qüestió depenen del que si li adjunti a continuació.

```

String ip = "https://clubtennisn.000webhostapp.com/bd_remota/";
String url = ip+"ClubTPConsulta.php?nick="+nick;
url=url.replaceAll( regex: " ", replacement: "%20");

final JSONObjectRequest getRequest = new JSONObjectRequest(Request.Method.GET, url, jsonRequest: null,
    new Response.Listener<JSONObject>()
{
    // ...
}

```

Il·lustració 67: Petició GET 1

El paràmetre de la *ip* apareix de color verd perquè s'està consultant el seu valor del arxiu *Strings* tal com s'ha explicat anteriorment. La URL és on s'especifica quina es la direcció completa on s'enviarà la petició. Per poder codificar la possibilitat d'espais en blanc a l'URL es substitueixen pel codi "%20".

Per aconseguir obtenir la informació demanada al servidor remot s'ha de crear un mètode *listener* per poder escoltar i capturar la resposta de la petició enviada. En el cas de la petició GET i tal com es pot veure a la imatge anterior en aquest cas *Volley* espera una resposta en format JSON.

Per poder enviar correctament una petició GET al servidor i gestionar la resposta es necessita la implementació de 3 mètodes per gestionar la petició correctament.

1. onResponse(JSONObject)

Aquesta funció es troba a dins del *listener* i és l'encarregada de rebre la resposta en format JSON i poder desglossar-la per obtenir la informació que ens interessa de manera ordenada.

```

@Override
public void onResponse(JSONObject response) {
    Log.d(tag: "Response", response.toString());
    try {
        JSONArray recs = response.getJSONArray( name: "user");
        JSONObject rec = recs.getJSONObject( index: 0);
        String nick_r = rec.getString( name: "nick");
        String email = rec.getString( name: "email");
    }
}

```

II-lustració 68: Petició GET 2

En aquest exemple es mostra que la funció espera una resposta en format JSON de tipus “user” en aquest cas que conté la informació del sobrenom i del correu electrònic de l'usuari.

2. onErrorResponse(VolleyError)

Aquesta funció es troba dins de l'*error listener* per si hi ha agut algun problema en la comunicació amb el servidor. Si es produeix algun error, s'executa i es mostra quin error ha sigut gràcies al *Log* (eina d'*Android* permet als programadors consultar dades concretes pel terminal amagades a l'usuari final). A més a més, en aquest cas es mostra un missatge a l'usuari de que hi ha hagut un problema amb el servidor.

```

new Response.ErrorListener()
{
    @Override
    public void onErrorResponse(VolleyError error) {
        Log.d(tag: "Error.Response", error.toString());
        progreso.dismiss();
        Toast.makeText(getApplicationContext(), text: "Problema amb el servidor", Toast.LENGTH_LONG).show();
    }
}

```

II-lustració 69: Petició GET 3

3. getHeaders()

Aquesta funció permet afegir una autenticació a les peticions en format d'usuari i contrasenya que autentifica el servidor remot i permet una connexió segura. Tant l'usuari com la contrasenya estan codificats en Base64.

```

@Override
public Map<String, String> getHeaders() {
    Map<String, String> headers = new HashMap<>();
    String usuari=getString(R.string.user);
    String contrasenya=getString(R.string.pwd);
    String credentials=usuari+" "+contrasenya;
    String auth = "Basic "
        + Base64.encodeToString(credentials.getBytes(), Base64.NO_WRAP);
    headers.put( k: "Authorization", auth);
    return headers;
}

```

II-lustració 70: Petició GET 4

Petició POST

Primer de tot s'inicialitza la petició POST especificant l'URL del servidor remot.

```

String ip = "https://clubtennisn.000webhostapp.com/bd_remota/";
String URL = ip+"ClubTFModifica.php?";
StringRequest getRequest = new StringRequest(Request.Method.POST, URL, new Response.Listener<String>() {
}

```

II-lustració 71: Petició POST 1

El paràmetre de la *ip* apareix de color verd perquè s'està consultant el seu valor de l'arxiu *Strings* tal com s'ha explicat anteriorment.

Per aconseguir obtenir la informació demanada al servidor remot s'ha de crear un mètode *listener* per poder escoltar i capturar la resposta de la petició enviada. En el cas de la petició POST i tal com es pot veure a la imatge anterior en aquest cas *Volley* espera una resposta en format *String*.

Això es deu al fet que com aquestes peticions provoquen modificacions a les taules només hem de rebre si s'ha completat la modificació correctament amb una cadena de caràcters.

Per poder enviar correctament una petició POST al servidor i gestionar la resposta es necessita la implementació de 4 mètodes per gestionar la petició correctament.

1. onResponse(String)

Aquesta funció es troba a dins del *listener* i és l'encarregada de rebre la resposta en format *String* per poder veure si la modificació s'ha completat correctament.

```
@Override
public void onResponse(String response) {
    Log.i(tag: "VOLLEY", response);

    switch (response){
        case "no_ok":
            progreso.dismiss();
            Toast.makeText(getApplicationContext(), text: "Contrasenya incorrecta!", Toast.LENGTH_LONG).show();
            break;
        case "ok_status0":
            progreso.dismiss();
            Toast.makeText(getApplicationContext(), text: "Activa primer la compta pel correu electrònic.", Toast.LENGTH_LONG).show();
            break;
        case "ok_status1":
            get_info_usuari(nickname, contrasenya);
            break;
        default:
            progreso.dismiss();
            Toast.makeText(getApplicationContext(), text: "Error: "+response, Toast.LENGTH_LONG).show();
            break;
    }
}
```

Il·lustració 72: Petició POST 2

En aquest exemple es mostra que depenen de la cadena de caràcters es porta a terme una acció o un altre.

2. onErrorResponse(VolleyError)

Aquesta funció es troba dins de l'*error listener* per si hi ha agut algun problema en la comunicació amb el servidor i funciona exactament igual que amb la petició GET.

```
new Response.ErrorListener()
{
    @Override
    public void onErrorResponse(VolleyError error) {
        Log.d(tag: "Error.Response", error.toString());
        progreso.dismiss();
        Toast.makeText(getApplicationContext(), text: "Problema amb el servidor", Toast.LENGTH_LONG).show();
    }
}
```

Il·lustració 73: Petició POST 3

3. getHeaders()

Aquesta funció permet afegir una autenticació a les peticions en format d'usuari i contrasenya que autentifica el servidor remot i permet una connexió segura. Tant l'usuari com la contrasenya estan codificats en Base64. Funciona exactament igual que amb la petició GET.

```
@Override
public Map<String, String> getHeaders() {
    Map<String, String> headers = new HashMap<>();
    String usuari=getString(R.string.user);
    String contrasenya=getString(R.string.pwd);
    String credentials=usuari+"."+contrasenya;
    String auth = "Basic "
        + Base64.encodeToString(credentials.getBytes(), Base64.NO_WRAP);
    headers.put("Authorization", auth);
    return headers;
}
```

Il·lustració 74: Petició POST 4

4. getParams()

Per enviar les dades mitjançant una petició POST necessitem un mètode més a diferència de les peticions GET que insereixen les dades directament a l'URL.

```
@Override
protected Map<String, String> getParams() {
    Map<String, String> param= new HashMap<>();
    param.put( k: "nickL", nickname);
    param.put( k: "pwdL", contrasenya);
    return param;
}
```

Il·lustració 75: Petició POST 5

Aquesta funció permet emmagatzemar les dades en format clau-valor i retorna un objecte *Volley*.

5.1.9. Shared preferences

Aquesta funcionalitat permet emmagatzemar informació en els dispositius mòbils en format clau-valor, és a dir, es pot definir una clau amb una cadena de caràcters i associar-hi qualsevol informació. Lo interessant es que aquesta informació es guarda en el dispositiu encara que tanquis l'aplicació.

En el cas d'aquesta aplicació l'objectiu d'utilitzar aquesta funcionalitat és que l'inici de sessió sigui automàtic en tornar a entrar a l'aplicació quan ja s'ha iniciat sessió anteriorment. D'aquesta manera no cal haver de posar el *nick* i la contrasenya cada cop.

```
public void emmagatzema_shared_preferences(String nick,String contrasenya){
    SharedPreferences.Editor editor = getSharedPreferences(PREFS_NICK, MODE_PRIVATE).edit();
    editor.putString( S: "user_nick",nick);
    editor.putString( S: "user_pwd",contrasenya);
    editor.apply();
}
```

Il·lustració 76: Emmagatzematge del Shared Preferences

5.1.10. Tasques asíncrones

Les tasques asíncrones és una eina que s'utilitza molt en aquesta aplicació. Com que totes les tasques s'executen en un fil principal (*main thread*) el problema sorgeix quan es volen executar varies tasques que tarden un temps a finalitzar i això pot provocar que l'aplicació es pugui bloquejar. Per altra banda, gràcies a les tasques asíncrones hi pot haver un paral·lelisme entre diferents tasques.

Dins del projecte s'utilitzen les *AsyncTask* per iniciar una petició i poder rebre la resposta del servidor remot.

```
public void ClassCanviarUsuari(String nom, String pwd){
    new Canviar_Usuari().execute(nom,pwd);
}
@SuppressWarnings("StaticFieldLeak")
private class Canviar_Usuari extends AsyncTask<String, Void, Void>{
    protected void onPreExecute() {
        progreso = new ProgressDialog( context: MainActivity.this);
        progreso.setMessage("Carregant...");
        progreso.show();
        progreso.setCancelable(false);
    }
    protected Void doInBackground(String... inf) {
        canviar_usuari(inf[0],inf[1]);
        return null;
    }
}
```

Il·lustració 77: Implementació de AsyncTask

Per iniciar una tasca asíncrona s'utilitza la funcionalitat *execute()*, que permet el pas de paràmetres per executar una funció de manera asíncrona. D'aquesta funcionalitat s'ha de destacar el *onPreExecute()* que s'executa abans que el mètode principal *doInBackground()* i

permet obrir un *progres dialog* que mostra a l'usuari un quadre de càrrega perquè esperi la resolució de la funció. La funció *doInBackground()* és on s'executen les funcions necessàries del fil que s'executa paral·lelament amb el fil principal. Finalment, en aquest cas no ha fet falta però, també es pot utilitzar la funció *onPostExecute()* si interessa executar alguna funció quan acaba la taca asíncrona.

5.2. Part del servidor

5.2.1. Gestió de la base de dades remota

El servei web *000webhost* disposa de la gestió directa de la base de dades per poder fer algun canvi molt concret si és necessari. A l'accedir-hi es necessita un usuari de la base de dades i la seva contrasenya per realitzar la connexió. Des de la interfície de *000webhost* es pot crear una base de dades *MySQL* i s'hi pot accedir utilitzant *phpMyAdmin*.

DB Name	DB User	DB Host	
id6206731_tpcclub	id6206731_xemnas	localhost	<div>Manage ▾</div> <div> <div>PhpMyAdmin</div> <div>Change Password</div> <div>Delete DB</div> <div> <div>DB Size: 7 MB</div> <div>DB Tables: 11</div> </div> </div>

Il·lustració 78: Gestió de la base de dades des de *000webhost*

Un cop s'entra al *PhpMyAdmin* amb la contrasenya i l'usuari de la taula que toca es pot observar un panel de control on es veuen totes les taules i es pot accedir a elles per modificar-les o consultar-les. A més a més, *phpMyAdmin* permet veure quina codificació de paràmetres (UTF-8) i quin motor de la base de dades s'està utilitzant. També es pot observar el pes de les taules i el nombre de files de cada taula.

Il·lustració 79: Menú de control de *phpMyAdmin*

5.2.2. Fitxers del servidor

Els fitxers del servidor programats amb PHP es poden dividir en dos grups diferenciats. Els encarregats de rebre peticions externes des de l'aplicació i els que reben les peticions a través d'enllaços d'activació, de canvi d'informació, de canvi de contrasenya o del programa de *pyhton* (que registra el temps atmosfèric). Els primers es troben a dins del servidor públic al directori de "bd_remota" i els segons al directori de "control_intern".

Dins dels fitxers PHP primer de tot s'inicialitza la connexió amb la base de dades per poder gestionar-la utilitzant la funcionalitat de "MySQLi_connect" posant abans de tot el nom del host, el nom de la base de dades, el nom d'usuari i la contrasenya per accedir-hi.

```
$hostname_localhost="localhost";
$databse_localhost="id6206731_tpcclub";
$username_localhost="id6206731_xemnas";
$password_localhost="";

include 'SendMail.php';
$base_url="https://clubtennisn.000webhostapp.com/bd_remota/";
$base_url_interna="https://clubtennisn.000webhostapp.com/control_intern/";

$conexio=mysqli_connect($hostname_localhost,$username_localhost,$password_localhost,$databse_localhost);
```

Il·lustració 80: Connexió amb la base de dades

També es pot observar com s'inclou el fitxer que conté la funció per enviar correus i les dues URL possibles com a variables globals per tal que es puguin modificar fàcilment des de dalt del fitxer, ja que s'utilitzen a molts llocs diferents.

A continuació, depenen de les capçaleres de les peticions, es tria una petició o una altra d'una llista de condicionals i es filtren per evitar injeccions de SQL tal com ja s'ha explicat a l'apartat de [disseny de seguretat del servidor](#).

Un cop filtrades les peticions es poden executar les consultes o modificacions amb els paràmetres totalment depurats.

```
elseif(isset($_GET["Tusers"])) {  
    $select="select * from user";  
    $resultat_select=mysqli_query($conexio,$select);  
    ...  
}
```

Il·lustració 81: Sentència SQL

Per retornar la informació en format JSON només s'ha d'executar la funció `json_encode` on la variable `$json` és un `array`.

```
while($registre=mysqli_fetch_array($resultat_select)){  
    $result["nick"]=$registre["nick"];  
    $result["name"]=$registre["name"];  
    $result["surname"]=$registre["surname"];  
    $result["email"]=$registre["email"];  
    $result["phone"]=$registre["phone"];  
    $result["sexe"]=$registre["sexe"];  
    $result["dni"]=$registre["dni"];  
    $result["imatge"]=base64_encode($registre["imatge"]);  
    ...  
    $json['user'][]=$result;  
    ...  
    echo json_encode($json);  
    mysqli_close($conexio);  
}
```

Il·lustració 82: Enviament de la resposta en format JSON

En el cas de les consultes s'utilitza la funció `MySQLi_fetch_array` per obtenir les columnes i la seva informació de les possibles consultes. En el cas de les modificacions s'ha utilitzat la funció `MySQLi_affected_rows` que retorna el nombre de columnes modificades després de l'última sentència SQL.

```
$update="update team set team.match= '{$num_match}' where team.name = '{$name}'";  
$resultat_update=mysqli_query($conexio,$update);  
$affected_rows=mysqli_affected_rows($conexio);  
if($affected_rows>0){  
    $resposta="ok";  
}  
else{  
    $resposta="no_ok";  
}
```

Il·lustració 83: Exemple de modificació

5.2.3. Enviament de correus amb *PHPMailer*

L'encarregat d'enviar els correus electrònics als usuaris és el correu del club, el qual si accedeix de manera automàtica i remota des del servidor utilitzant la llibreria *PHPMailer*. Utilitzant el fitxer "Send_Mail.php" es crida una funció que utilitza la llibreria *PHPMailer* per crear un objecte *PHPMailer()*. Aquest mòdul consta d'una sola funció que varia segons el títol i el missatge que s'hi introdueix i s'utilitza per a totes les interaccions amb els correus electrònics dels usuaris.

```
use PHPMailer\PHPMailer\PHPMailer;  
use PHPMailer\PHPMailer\Exception;  
  
require 'PHPMailer/src/Exception.php';  
require 'PHPMailer/src/PHPMailer.php';  
require 'PHPMailer/src/SMTP.php';
```

```

function Send_Mail($to,$subject,$body){
    $from      = "clubtennis@navas.cat";
    $mail      = new PHPMailer();
    $mail->CharSet = "UTF-8";
    $mail->IsSMTP(true);           // use SMTP
    $mail->IsHTML(true);
    $mail->SMTPAuth = true;         // enable SMTP authentication
    // $mail->Host      = "smtp.gmail.com"; // "tls://smtp.yourwebsite.com"; // SMTP host
    $mail->SMTPSecure = 'ssl';
    $mail->Host = 'smtp.gmail.com';
    $mail->Port = 465; // 465;           // set the SMTP port
    $mail->Username = "clubtennis@gmail.com"; // SMTP username
    $mail->Password = "XXXXXXXXXX"; // SMTP password
    $mail->SetFrom($from, 'Club Tennis Navàs');
    $mail->AddReplyTo($from, 'Club Tennis Navàs');
    $mail->Subject = $subject;
    $mail->MsgHTML($body);
    $address = $to;
    $mail->AddAddress($address, $to);
    $exit = $mail->Send();

    if($exit){ echo 'Correu enviat correctament.'; }
    else{ echo "Error: " . $phpmailer->ErrorInfo; }
}

```

Il·lustració 84: Codi per enviar correus electrònics des del servidor

5.3. Part externa

5.3.1. Programa pyhton de registre atmosfèric

Aquest programa esta dividit en 3 funcions principals:

1) temps_actual_club()

Aquesta funció és l'encarregada de vincular-se amb la pàgina web d'*OpenWeather Map* utilitzant un clau registrant-se de manera gratuïta per tal de poder consultar el temps atmosfèric. Quan es crida aquesta funció es registra el temps actual a les pistes de tennis de Navàs i es guarda la informació en format clau valor a dins d'un diccionari de *pyhton*. Cada clau del diccionari és una característica atmosfèrica. Finalment, un cop s'ha registrat la informació al diccionari aquesta funció retorna el diccionari.

```

def temps_actual_club():
    #Inicialització d'un diccionari buit
    d_inf={}

    #Clau per utilitzar l'API.
    apikey = "041cc267f37bdaf8c2efcd542d43c675"

    #Es crea una instància OWM pasant la clau per l'us de la API.
    owm = pyowm.OWM(apikey)

    #Es defineix la ciutat per nom o es passen les coordenades.
    #Pista tennis Navas
    obs = owm.weather_at_coords(41.900173,1.882256)
    #obs = owm.weather_at_place('Navas,ES')

    #S'instancien les dades de l'estació meteorologica.
    w = obs.get_weather()

    print "Dia i hora: ",w.get_reference_time(timeformat='iso')
    d_inf["Reference time"]=w.get_reference_time(timeformat='iso')

    print "Núvols: ", w.get_clouds()
    d_inf["Clouds"]=w.get_clouds()

    print "Plujes: ", w.get_rain()

```

Il·lustració 85: Funció del temps actual

2) petició_post(diccionari)

Aquesta funció rep el diccionari amb la informació atmosfèrica i l'envia cap al servidor remot amb una petició POST seguin l'estructura clau-valor amb la qual venia al diccionari.


```
def peticio_post(d):
    r = requests.post("https://clubtennisn.000webhostapp.com/control_intern/temps.php", data={'reference_time': d['Reference time'][:3], 'clouds': d['Clouds'], 'rain': d['Rain'], 'snow': d['Snow'], 'wind': d['Wind']['speed'], 'humidity': d['Humidity'], 'pressure': d['Pressure']['press'], 'temperature': d['Temperature']['temp'], 'status': d['Status'], 'detailed_status': d['Detailed status'], 'sunrise_time': d['Sunrise time'][:3], 'sunset_time': d['Sunset time'][:3]})

    print(r.status_code, r.reason)

    print(r.text[:300])
```

Il·lustració 86: Funció de la petició post

3) crida_hores()

És la funció encarregada d'executar les altres dos quan sigui l'hora que toca. L'hora es controla comprovant l'hora actual en tot moment i llençant una petició a cada hora en el minut especificat a la variable "minuts". En aquest cas, la variable "minuts" té el valor 30, pel que s'envia la informació dels temps al minut 30 de cada hora.

```
def crida_hores():
    minuts=30
    ultima_hora=0
    while (True):
        t=time.localtime()
        h=t.tm_hour
        m=t.tm_min
        s=t.tm_sec
        if (h==7 and m==minuts and h!=ultima_hora):
            d=temps_actual_club()
            peticio_post(d)
            ultima_hora=h
            print 'PETICIÓ ENVIADA A LES '+str(h)+'h i'+str(m)+'min'

        elif (h==8 and m==minuts and h!=ultima_hora):
            d=temps_actual_club()
            peticio_post(d)
            ultima_hora=h
            print 'PETICIÓ ENVIADA A LES '+str(h)+'h i'+str(m)+'min'
```

Il·lustració 87: Funció de la crida d'hores

En el *main* del programa només s'executa la funció *crida_hores()* directament, ja que aquesta és l'encarregada d'anar consultant l'hora i executar les altres dues funcions quan toqui.

6. Proves

Aquest projecte ha tingut proves constants en tot moment a mesura que s'anaven afegint noves funcionalitats per comprovar el correcte funcionament d'aquestes en tot moment. A continuació s'expliquen les proves i els casos d'accions que pot realitzar un usuari, com es tracten i quina és la seva resposta esperada.

En el segon punt d'aquest apartat s'inclou una petita explicació de les versions que s'han pogut provar i validar per assegurar el funcionament correcte de l'aplicació.

6.1. Casos de prova

Els casos de prova serveixen per veure com reaccionen les diferents funcionalitats disponibles d'un usuari. Aquestes funcionalitats esperen uns valors d'entrada per iniciar un procés que dona un resultat o un error i el mostra per pantalla.

6.1.1. Casos de prova per usuaris normals

Per dur a terme aquests casos de prova s'han seleccionat les principals funcionalitats de l'aplicació que utilitzen els usuaris normals aquí llistades:

- Registre
- Inici de sessió
- Editar perfil

- Recuperar contrasenya
- Donar de baixa un usuari
- Reservar pista
- Eliminar reserva
- Registre a un torneig
- Modificar partit

En cada un dels casos s'especifica quin és el propòsit, els requisits previs, les variables d'entrada, el fil de l'execució i els possibles resultats.

CAS DE PROVA		REGISTRE
Propòsit	Registrar un usuari nou.	
Requisits previs	L'usuari ha de voler unir-se al club i no estar a la llista negra.	
Variable d'entrada	<ul style="list-style-type: none"> ▪ Sobrenom ▪ Nom ▪ Cognom ▪ Correu electrònic ▪ Telèfon ▪ Contrasenya ▪ Confirmar contrasenya ▪ DNI ▪ Sexe ▪ Imatge 	
Fil de l'execució	<ol style="list-style-type: none"> 1) Validar tots els paràmetres d'entrada. 2) Verificar si ja existeix el sobrenom, el correu electrònic, el telèfon o el DNI. 3) S'envia un correu electrònic per poder activar el compte. 4) Un cop el compte s'ha activat, es pot utilitzar amb normalitat. 	
Possibles resultats	<p>En superar-se els passos 1 i 2, es registra l'usuari i ja només s'ha d'activar el compte.</p> <p>En cas contrari, s'informa a l'usuari a quin pas hi ha el problema.</p>	

Taula 10: Cas de prova del registre

CAS DE PROVA		INICI DE SESSIÓ
Propòsit	Iniciar sessió d'un usuari.	
Requisits previs	L'usuari ha d'existir.	
Variable d'entrada	<ul style="list-style-type: none"> ▪ Sobrenom ▪ Contrasenya 	
Fil de l'execució	<ol style="list-style-type: none"> 1) Validació dels paràmetres d'entrada 2) Verificació de l'existència de l'usuari introduït. 3) Validació per mirar que la contrasenya sigui correcta tenint en compte el sobrenom introduït. 	
Possibles resultats	<p>En superar-se tots els passos s'inicia sessió.</p> <p>En cas contrari, s'informa a l'usuari a quin pas hi ha el problema.</p>	

Taula 11: Cas de prova del inici de sessió

CAS DE PROVA		EDITAR PERFIL
Propòsit	Actualitzar la informació d'un usuari	
Requisits previs	L'usuari ha d'existir i ha d'haver iniciat sessió.	
Variable d'entrada	<ul style="list-style-type: none"> ▪ Sobrenom ▪ Nom ▪ Cognom ▪ Correu electrònic ▪ Telèfon ▪ Contrasenya ▪ Confirmar contrasenya ▪ DNI ▪ Sexe ▪ Imatge 	
Fil de l'execució	<ol style="list-style-type: none"> 1) Validar tots els paràmetres d'entrada. 2) Verificar si ja existeix el nou sobrenom, el nou correu electrònic, el nou telèfon o el DNI. 3) S'envia correu electrònic per poder confirmar els canvis. 	
Possibles resultats	<p>En superar-se tots els passos s'envia el correu de confirmació i un cop confirmat es reflecteix en la informació.</p> <p>En cas contrari, s'informa a l'usuari a quin pas hi ha el problema.</p>	

Taula 12: Cas de prova per editar perfil

CAS DE PROVA		RECUPERAR CONTRASENYA
Propòsit	Aconseguir una contrasenya de recuperació.	
Requisits previs	L'usuari ha d'existir.	
Variable d'entrada	<ul style="list-style-type: none"> ▪ Sobrenom 	
Fil de l'execució	<ol style="list-style-type: none"> 1) Validar el sobrenom introduït. 2) Enviar contrasenya de recuperació i enllaç de confirmació al correu electrònic. 	
Possibles resultats	<p>Un cop validat el nom s'envia la contrasenya i si es confirma en el correu electrònic, es produeix el canvi.</p> <p>En cas contrari, s'informa a l'usuari a quin pas hi ha el problema.</p>	

Taula 13: Cas de prova per recuperar contrasenya

CAS DE PROVA		DONAR DE BAIXA UN COMPTE
Propòsit	Donar de baixa un compte d'usuari.	
Requisits previs	L'usuari ha d'existir i ha d'haver iniciat sessió.	
Variable d'entrada	<ul style="list-style-type: none"> ▪ Sobrenom 	
Fil de l'execució	<ol style="list-style-type: none"> 1) En tenir ja la sessió iniciada, ja se sap segur que l'usuari existeix. 2) S'envia un correu electrònic de confirmació. 	
Possibles resultats	<p>Un cop s'ha confirmat s'envia un correu de confirmació i s'elimina el compte.</p> <p>En cas contrari, s'informa a l'usuari a quin pas hi ha el problema.</p>	

Taula 14: Cas de prova per donar de baixa un compte

CAS DE PROVA		RESERVAR UN PISTA
Propòsit	Reservar una pista per un dia i hores concretes.	
Requisits previs	L'usuari ha d'existir i ha d'haver iniciat sessió.	
Variable d'entrada	<ul style="list-style-type: none"> ▪ Dia ▪ Pista ▪ Hores ▪ Sobrenom ▪ Correu electrònic 	

Fil de l'execució	<ol style="list-style-type: none"> 1) L'usuari selecciona un dia disponible del calendari (no es pot seleccionar un dia anterior a l'actual ni un dia més llunyà de 3 mesos). 2) L'usuari tria una pista i selecciona les hores que vulgui i que estiguin disponibles (no es poden seleccionar més de 4 hores en una mateixa pista un mateix dia).
Possibles resultats	<p>En superar-se tots els passos es confirma la reserva.</p> <p>En cas contrari, s'informa a l'usuari a quin pas hi ha el problema.</p>

Taula 15: Cas de prova per reservar una pista

CAS DE PROVA ELIMINAR UNA RESERVA	
Propòsit	Eliminar una reserva d'un dia i una pista concreta.
Requisits previs	L'usuari ha de tenir com a mínim una reserva.
Variable d'entrada	<ul style="list-style-type: none"> ▪ Dia ▪ Pista ▪ Sobrenom
Fil de l'execució	<ol style="list-style-type: none"> 1) L'usuari selecciona un dia disponible del calendari. 2) L'usuari tria una pista. 3) L'usuari tria eliminar la reserva d'aquella pista.
Possibles resultats	<p>En superar-se tots els passos es confirma l'eliminació de la reserva si n'hi ha una en la pista seleccionada.</p> <p>En cas contrari, s'informa a l'usuari a quin pas hi ha el problema.</p>

Taula 16: Cas de prova per eliminar una reserva

CAS DE PROVA REGISTRE A UN TORNEIG	
Propòsit	Permetre un registre a un torneig.
Requisits previs	<p>L'usuari no pot estar participant ja en el torneig en un altre equip.</p> <p>L'usuari ha de complir les especificacions del torneig.</p>
Variable d'entrada	<ul style="list-style-type: none"> ▪ Nom del torneig ▪ Sobrenom jugador 1 ▪ Sobrenom jugador 2 (només si el torneig és per parelles)
Fil de l'execució	<p>Aquest cas de prova té dos fils d'execució segons si el torneig és individual o per parelles:</p> <p>Individual</p> <ol style="list-style-type: none"> 1) Consultar la informació del torneig on es vol registrar. 2) Comprovar quants equips hi ha registrats. 3) Comprovar que el gènere de l'usuari compleix l'especificació del torneig actual. 4) Comprovar si el jugador ja participa en un altre equip o si el torneig ja està ple. 5) Comprovar que el nom d'equip no existeixi ja en el torneig. 6) Afegir l'equip individual de l'usuari actual al torneig. <p>Parelles</p> <ol style="list-style-type: none"> 1) Consultar la informació del torneig on es vol registrar. 2) Comprovar quants equips hi ha registrats. 3) Comprovar que el gènere de l'usuari (jugador 1) compleix l'especificació del torneig actual. 4) Comprovar si el jugador ja participa en un altre equip o si el torneig ja està ple. 5) Comprovar el sobrenom del segon integrant de l'equip per si no existeix en el club. 6) Comprovar que el gènere del segon integrant compleix l'especificació del torneig actual. 7) Comprovar si el segon integrant ja participa en el torneig en un altre equip. 8) Comprovar que el nom d'equip no existeixi ja en el torneig.

	9) Afegir l'equip per parelles de l'usuari actual i el seu company al torneig.
Possibles resultats	En superar-se tots els passos es confirma el registre de l'equip creat al torneig seleccionat per l'usuari actual. En cas contrari, s'informa a l'usuari a quin pas hi ha el problema.

Taula 17: Cas de prova del registre d'un torneig

CAS DE PROVA		MODIFICAR UN PARTIT
Propòsit		Poder modificar un partit.
Requisits previs		L'usuari ha de tenir un partit actiu en un torneig.
Variable d'entrada		<ul style="list-style-type: none"> ▪ Nom del torneig ▪ Si és victòria o derrota ▪ El resultat del partit ▪ Sobrenom de l'usuari
Fil de l'execució		<ol style="list-style-type: none"> 1) Consultar la informació del torneig. 2) Comprovar si l'usuari actual participa en el torneig i té un partit actiu. 3) Consultar la informació del partit actiu de l'usuari. 4) Comprovar si es vol registrar una victòria o una derrota. 5) Comprovar si es pot fer la modificació tenint en compte la informació del partit. 6) Actualitzar el partit.
Possibles resultats		En superar-se tots els passos es confirma la modificació en el partit actiu de l'usuari actual. En cas contrari, s'informa a l'usuari a quin pas hi ha el problema.

Taula 18: Cas de prova per modificar un partit

6.1.2. Casos de prova per administradors

Per dur a terme aquests casos de prova s'han seleccionat les principals funcionalitats de l'aplicació que utilitzen els administradors aquí llistades:

- Eliminar un usuari
- Ocupar hores de reserva
- Ocupar dies de reserva
- Treure un usuari de la llista negra
- Missatge de difusió
- Omplir torneig (igual que amb usuaris normals però ho controla un administrador)
- Buidar torneig
- Modificar partit (igual que amb usuaris normals però ho controla un administrador)

CAS DE PROVA		ELIMINAR UN USUARI
Propòsit		Expulsar un usuari del club per alguna raó.
Requisits previs		Que l'usuari estigui registrat al club i hagi comés alguna infracció greu.
Variable d'entrada		<ul style="list-style-type: none"> ▪ Sobrenom de l'usuari que es vol expulsar ▪ Raó de l'expulsió
Fil de l'execució		<ol style="list-style-type: none"> 1) Es tria un usuari disponible. 2) Es dóna una raó de l'expulsió. 3) S'introdueixen les dades importants de l'usuari a una llista negra (per controlar si es vol tornar a registrar). 4) S'elimina l'usuari seleccionat.

Possibles resultats	En superar-se tots els passos es confirma l'eliminació de l'usuari seleccionat. En cas contrari, s'informa a l'administrador a quin pas hi ha el problema.
----------------------------	---

Taula 19: Cas de prova per eliminar un usuari

CAS DE PROVA		OCUPAR HORES DE RESERVA
Propòsit		Possibilitat d'ocupar tant hores buides com ocupades.
Requisits previs		-
Variable d'entrada		<ul style="list-style-type: none"> ▪ Dia ▪ Pista ▪ Hores ▪ Sobrenom de l'administrador
Fil de l'execució		<ol style="list-style-type: none"> 1) S'envia la petició per ocupar les hores seleccionades. 2) Es comprova si estan ocupades per altres usuaris. 3) Si estan ocupades, s'envia un correu electrònic a les persones que tenien una reserva per informar quines hores se'ls hi ha ocupat. 4) Es confirma l'ocupació de les hores.
Possibles resultats		En superar-se tots els passos es confirma l'ocupació de les hores seleccionades per l'administrador. En cas contrari, s'informa a l'administrador a quin pas hi ha el problema.

Taula 20: Cas de prova per ocupar hores de reserva

CAS DE PROVA		OCUPAR DIES DE RESERVA
Propòsit		Possibilitat d'ocupar dies sencers.
Requisits previs		-
Variable d'entrada		<ul style="list-style-type: none"> ▪ Dia ▪ Sobrenom de l'administrador
Fil de l'execució		<ol style="list-style-type: none"> 1) S'envia la petició per ocupar totes les hores de totes pistes del dia seleccionat. 2) Es comprova si estan ocupades per altres usuaris. 3) Si estan ocupades, s'envia un correu electrònic a les persones que tenien una reserva per informar quines hores se'ls hi ha ocupat. 4) Es confirma l'ocupació del dia.
Possibles resultats		En superar-se tots els passos es confirma l'ocupació del dia seleccionat per l'administrador. En cas contrari, s'informa a l'administrador a quin pas hi ha el problema.

Taula 21: Cas de prova per ocupar dies de reserva

CAS DE PROVA		TREURE UN USUARI DE LA LLISTA NEGRA
Propòsit		Treure un usuari de la llista negra.
Requisits previs		Que hi hagi la informació d'algun usuari a la llista negra.
Variable d'entrada		<ul style="list-style-type: none"> ▪ Sobrenom antic de l'usuari expulsat
Fil de l'execució		<ol style="list-style-type: none"> 1) Se selecciona un usuari expulsat comprovant el seu DNI. 2) Es confirma l'operació de treure'l de la llista negra.
Possibles resultats		En superar-se tots els passos es confirma l'operació de treure l'usuari seleccionat de la llista negra. En cas contrari, s'informa a l'administrador a quin pas hi ha el problema.

Taula 22: Cas de prova per treure un usuari de la llista negra

CAS DE PROVA		ENVIAR MISSATGE DE DIFUSIÓ
Propòsit	Enviar un missatge de difusió a tots els usuaris normals o a tots els administradors.	
Requisits previs	-	
Variable d'entrada	<ul style="list-style-type: none"> ▪ El missatge de difusió ▪ Usuaris normals o administradors 	
Fil de l'execució	<ol style="list-style-type: none"> 1) Es tria si es vol enviar a tots els usuaris o a tots els administradors 2) S'envia la petició amb el missatge 3) En el servidor se seleccionen els usuaris que toquen i se'ls hi envia a tots el missatge al correu electrònic. 4) Es confirma que la difusió s'ha enviat. 	
Possibles resultats	<p>En superar-se tots els passos es confirma que la difusió s'ha enviat correctament al grup que toca.</p> <p>En cas contrari, s'informa a l'administrador a quin pas hi ha el problema.</p>	

Taula 23: Cas de prova del missatge de difusió

CAS DE PROVA		BUIDAR TORNEIG
Propòsit	Treure equips quan el torneig encara s'està omplint, és a dir, que està creat però no actiu.	
Requisits previs	Que hi hagi un torneig creat però no actiu i es vulgui treure un equip.	
Variable d'entrada	<ul style="list-style-type: none"> ▪ Nom de l'equip 	
Fil de l'execució	<ol style="list-style-type: none"> 1) Se selecciona el torneig 2) Es comprova que l'equip està a dins del torneig 3) Es confirma l'expulsió de l'equip en el torneig seleccionat 	
Possibles resultats	<p>En superar-se tots els passos es confirma l'expulsió de l'equip en el torneig seleccionat per l'usuari</p> <p>En cas contrari, s'informa a l'administrador a quin pas hi ha el problema.</p>	

Taula 24: Cas de prova per buidar un torneig

7. Pressupost de l'aplicació

En aquest apartat es mostra el possible valor de l'aplicació en cas que es busques la seva comercialització de manera professional. Com a autor del projecte he assumit les diferents funcions d'analista, programador i he sigut l'encarregat de documentar la informació del projecte.

En un projecte real aquests 3 rols estan molt ben diferenciats i els seus treballadors reben remuneracions diferents. Per fer un càlcul estimat dels pressupostos de l'aplicació, s'utilitzen els següents preus de mà d'obra:

- **Analista:** realitza les funcions de recopilació d'informació, s'encarrega de l'anàlisi i del disseny. El preu-hora seria de 13 €/h.
- **Programador:** realitza la implementació del projecte i du a terme les diferents proves i testos que se li fan per assegurar la seva robustesa. El preu-hora seria de 14 €/h.
- **Escriptor:** realitza la recopilació de totes les funcionalitats del projecte i les documenta en una memòria. El preu-hora seria 12 €/h.

El projecte es va iniciar a finals de gener i principis de febrer de 2018 (s'estima el dia 1 de febrer) hi ha finalitzat a mig setembre (s'estima el dia 15 de setembre). Durant aquest temps s'han realitzat una mitjana de 36 hores cada setmana.

FEINES FETES	SETMANES TREBALLADES	HORES TREBALLADES	ENCARREGAT	COST (€)
Recopilar info.	1 setmanes	36 hores	Analista	468 €
Anàlisi	3 setmanes	72 hores	Analista	936 €
Disseny	4 setmanes	144 hores	Analista	1872 €
Implementació	16 setmanes	576 hores	Programador	8.064 €
Proves	3 setmanes	108 hores	Programador	1512 €
Documentació	3 setmanes	108 hores	Escriptor	1296 €
TOTAL	30 setmanes	1044 hores		14148 €

Taula 25: Cost de la mà d'obra

Tenint en compte les hores que treballa cada encarregat i el seu rol dins del projecte es pot calcular que el pressupost total de mà d'obra és 14148 €.

També s'ha de tenir en compte el pressupost del software i el hardware utilitzats. Els elements utilitzats de software són els següents:

SOFTWARE	COST (€)
OpenWeatherMap	0 € (Gratuït)
Android Studio i SDK	0 € (Gratuït)
XAMPP	0 € (Gratuït)
000webhost	0 € (Gratuït)
MySQL	0 € (Gratuït)
Llicència Windows 10 pro	45 €
Llicència Ubuntu 16.04	0 € (Gratuït)
Llicència SQLyog Free	0 € (Gratuït)
TOTAL	45 €

Taula 26: Cost del software

El cost relacionat amb el software és de 45 €. Els elements utilitzats de hardware són els següents:

HARDWARE	COST (€)
Ordinador sobretaula	400 €
Ordinador portàtil	300 €
Connexió a internet	240 €
Dispositiu Android	200 €
TOTAL	1140 €

Taula 27: Cost del hardware

El pressupost total per desenvolupar l'aplicació surt de la suma del cost de mà d'obra, el software i el hardware.

PART DEL PROJECTE	COST
Mà d'obra	14148 €
Software	45 €
Hardware	1140 €
TOTAL	15333 €

Taula 28: Cost total del projecte

8. Conclusions

Aquest projecte ha suposat un petit repte per mi, ja que no havia tocat mai el llenguatge Java ni PHP tan directament ni tampoc havia fet mai cap aplicació *Android* des de zero. La realització del projecte m'ha ajudat a veure més bé com funcionen les aplicacions *Android* i quines funcionalitats tenen disponibles per oferir una navegació fluida pels usuaris al llarg de l'aplicació.

Ha sigut molt interessant conèixer la pàgina web 000wehost que permet crear un servidor web amb base de dades de manera totalment gratuïta i sense gaires limitacions. A més a més, s'ha tingut en compte la seguretat de la transmissió d'informació aplicant diferents mètodes molt interessants d'aprendre.

Com que també volia programar alguna cosa amb *python*, he creat un petit programa que interactua amb una pàgina web meteorològica i el meu servidor web remot.

També he trobat molt interessant la llibreria *PHPMailer*, que m'ha permès poder enviar correus electrònics de manera totalment automàtica als usuaris que m'interessi des del servidor per a poder suplir les diferents funcionalitats de l'aplicació.

La conclusió final és que ha sigut molt interessant per les noves tecnologies que he après i que estic molt satisfet amb el resultat final de la primera versió de l'aplicació. Aquesta primera versió encara no està pública però la idea és que es publiqui en breus a la Play Store d'*Android*.

9. Treball futur

La primera idea com a treball futur és implementar també els costos de reserva depenent de les hores reservades per a cada usuari. També seria interessant portar les funcionalitats principals de l'aplicació a una pàgina web per tal de poder controlar-ho també tot des d'un ordinador.

Tot i que en alguns apartats és una aplicació molt genèrica que es podria extrapolar a diferents clubs en d'altres està més limitada per adaptar-se al club de tennis Navàs. Trobaria molt interessant en un futur, aconseguir que totes les funcionalitats fossin completament genèriques, de manera que es poguessin adaptar fàcilment a qualsevol club, independentment de les dimensions i el nombre de pistes d'aquest.

Una millora de futur important també seria la migració del servei web gratuït a un completament privat sense les limitacions que té actualment.

Com ja s'ha especificat al final de les conclusions, l'aplicació es penjarà a la Play Store d'*Android* per fer-se totalment pública en acabar el treball.

10. Referències utilitzades

10.1. Il·lustracions extretes d'internet

- 1) **Diagrama de l'ús de versions Android.** (en línia, 2018)
https://upload.wikimedia.org/wikipedia/commons/e/ee/Android_historical_version_distribution_-_vector.svg
- 2) **Aplicació “Coge Pista”.** (en línia, 2018)
https://upload.wikimedia.org/wikipedia/commons/e/ee/Android_historical_version_distribution_-_vector.svg
- 3) **Aplicació del “Club de Tenis”.** (en línia, 2018)
<https://play.google.com/store/apps/details?id=com.orointernet.chamartin>
- 4) **Aplicació “Playtomic”.** (en línia, 2018)
<https://play.google.com/store/apps/details?id=com.playtomic>
- 5) **Parts diferenciades del algoritme de xifratge Bcrypt.** (en línia, 2018)
<http://php.net/manual/en/faq.passwords.php>
- 6) **Logotip d'Android Studio.** (en línia, 2018)
<https://www.androidauthority.com/android-studio-tutorial-beginners-637572/>
- 7) **Logotip de XAMPP.** (en línia, 2018)
<https://mantenimientosdeunapc.blogspot.com/2011/11/que-es-xampp-y-para-que-sirve.html>
- 8) **Logotip d'Apache.** (en línia, 2018)
<https://nosturi.es/2017/10/10/desarrollo-funcionalidades-software-libre/apache-http-server/>
- 9) **Logotip de SQLyog.** (en línia, 2018)
<http://coding-cheese.blogspot.com/2017/01/sqlyog-1209-ultimate-full-serial.html>
- 10) **Logotip de MySQL.** (en línia, 2018)
<https://fireosoft.com.co/blogs/que-es-y-para-que-sirve-mysql/>
- 11) **Logotip de 000webhost.** (en línia, 2018)
<https://es.000webhost.com/>
- 12) **Logotip de PHPMailer.** (en línia, 2018)
<https://github.com/PHPMailer/PHPMailer>
- 13) **Logotip de la llibreria pyown.** (en línia, 2018)
<https://raw.githubusercontent.com/csparpa/pyowm/master/logos/180x180.png>
- 14) **Logotip de OpenWeatherMap.** (en línia, 2018)
<https://openweathermap.org/>
- 15) **Diagrama d'estats importants d'una activitat.** (en línia, 2018)
<https://developer.android.com/reference/android/app/Activity#ActivityLifecycle>
- 16) **Cicle de vida d'un fragment.** (en línia, 2018)

<https://developer.android.com/guide/components/fragments?hl=es-419>

17) **Esquema d'una activitat que conté 2 fragments.** (en línia, 2018)

<https://developer.android.com/guide/components/fragments?hl=es-419>

10.2.Referències web

18) **PHP password.** (en línia, 2018)

<http://php.net/manual/en/faq.passwords.php>

19) **PHP connexió.** (en línia, 2018)

https://www.w3schools.com/php/php_mysql_prepared_statements.asp

20) **Tutorial d'Android.** (en línia , 2018)

<https://www.androidauthority.com/android-studio-tutorial-beginners-637572/>

21) **Android Studio.** (en línia, 2018)

<https://developer.android.com/studio/>

22) **Xampp.** (en línia, 2018)

<https://www.apachefriends.org/es/index.html>

23) **Webhost000.** (en línia, 2018)

<https://es.000webhost.com/>

24) **SQLyog.** (en línia, 2018)

<https://www.webyog.com/product/sqlyog>

25) **Llibreria PHPMailer.** (en línia, 2018)

<https://github.com/PHPMailer/PHPMailer>

26) **Llibreria Volley.** (en línia, 2018)

<https://github.com/google/volley>

27) **Llibreria Pyowm.** (en línia, 2018)

<https://github.com/csparpa/pyowm>

28) **OpenWeatherMap.** (en línia, 2018)

<https://openweathermap.org/>

29) **MySQL.** (en línia, 2018)

<https://www.mysql.com/>

30) **StackOverflow.** (en línia, 2018)

<https://stackoverflow.com/>

31) **Java.** (en línia, 2018)

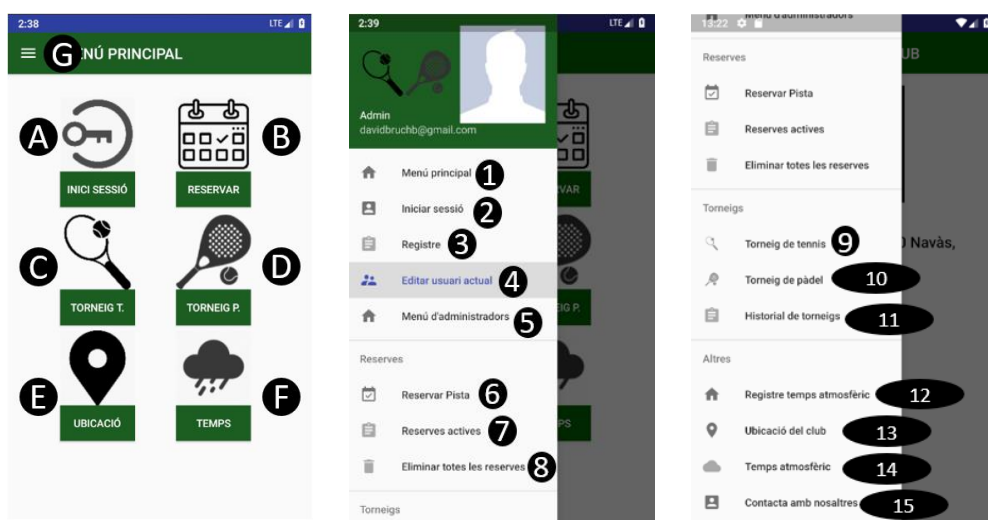
<https://docs.oracle.com/javase/tutorial/>

Annex A. Manual d'ús de l'aplicació

En aquest apartat es mostraran totes les pantalles dissenyades per aquesta aplicació i s'explicaran totes les seves funcionalitats possibles que pot dur a terme un usuari.

1. Menú principal i menú lateral

Quan l'usuari obre l'aplicació el primer que es troba és el menú principal amb les seves 6 opcions diferenciades i el menú desplegable amb les seves 14 opcions al costat, que estarà present i disponible a totes les pantalles que es mostraran més endavant. Si l'usuari no havia iniciat sessió abans amb anterioritat, tot i que pot veure totes les opcions, només pot entrar a l'opció d'inici de sessió i de registre.



Il·lustració 88: Menú principal i menú desplegable lateral

Interaccions disponibles al menú principal:

- Botó inici de sessió (A): obre la pantalla d'inici de sessió.
- Botó de reserva (B): obre el calendari per triar el dia de la reserva.
- Botó de torneig de tennis (C): obre la pantalla principal del torneig de tennis.
- Botó del torneig de pàdel (D): obre la pantalla principal del torneig de tennis.
- Botó d'ubicació (E): obre el *GoogleMaps* dins de l'aplicació amb la ubicació del club marcada.
- Botó del temps (F): obre el *Meteocat* dins de l'aplicació per veure el temps atmosfèric al club.
- Botó del menú desplegable (G): desplega el menú lateral per fer-lo visible. Aquest botó serà visible a totes les pantalles de l'aplicació i per aquesta raó no es tornarà a explicar.

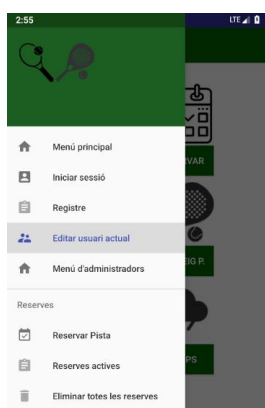
En tots els botons del menú principal cal destacar que es pot clicar tant el botó com la mateixa imatge que té a sobre per realitzar l'acció.

Interaccions disponibles al menú lateral:

- Botó del menú principal (1): obre el menú principal.
- Botó d'inici de sessió (2): obre la pantalla que permet iniciar sessió.
- Botó de registre (3): obre la pantalla que permet el registre.
- Botó d'editar usuari actual (4): obre la pantalla que permet editar la informació de l'usuari.
- Botó del menú d'administradors (5): obre el menú d'administradors i només hi poden accedir els administradors. Si un usuari regular intenta entrar-hi, li apareix un missatge d'avís explicant que no té autorització.
- Botó de reservar pista (6): obre el calendari per triar el dia de la reserva.
- Botó de reserves actives (7): obre una pantalla amb la llista de reserves actives de l'usuari. Si l'usuari no té cap reserva activa, apareix un missatge remarcant-ho.

- Botó per eliminar totes les reserves actives (8): permet eliminar totes les reserves actives de l'usuari actual.
- Botó del torneig de tennis (9): obre la pantalla per accedir al torneig de tennis actual.
- Botó del torneig de pàdel (10): obre la pantalla per accedir al torneig de pàdel actual.
- Botó de l'historial de torneigs (11): obre una pantalla per veure la informació dels torneigs ja finalitzats anteriorment.
- Botó del registre de temps atmosfèric (12): obre una pantalla que permet veure la informació del temps registrada.
- Botó d'ubicació del club (13): obre una pantalla que consulta *GoogleMaps* per mostrar on es troba el club.
- Botó del temps atmosfèric (14): obre una pantalla que consulta *Meteocat* per mostrar el temps en el club.
- Botó per contactar amb el club (15): obre una pantalla amb la informació de contacte del club.

Si l'usuari intenta entrar a qualsevol altra opció sense haver iniciat sessió abans li apareix un missatge d'avís que li diu que ha d'iniciar sessió o registrar-se abans per poder utilitzar les altres funcionalitats de l'aplicació. A més a més no l'hi apareix cap usuari i cap imatge d'usuari quan desplega el menú lateral per indicar que no s'ha iniciat sessió.



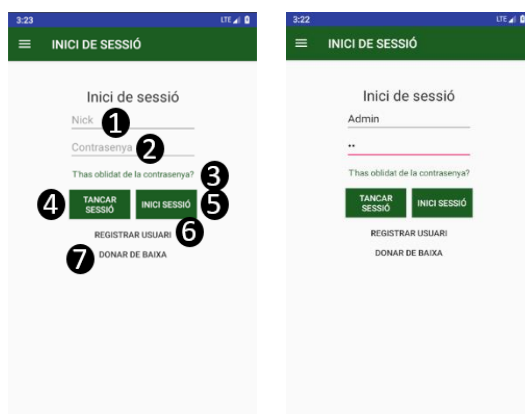
Il·lustració 90: Menú desplegable sens usuari



Il·lustració 89: Missatge d'avís per iniciar sessió o registrar

2. Inici de sessió

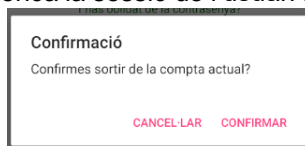
A la pantalla d'inici de sessió hi ha 2 espais per entrar informació, en el primer espai s'ha d'entrar el "nick" de l'usuari i en el segon la seva contrasenya tal com s'indica.



Il·lustració 91: Pantalla d'inici de sessió buida i omplerta

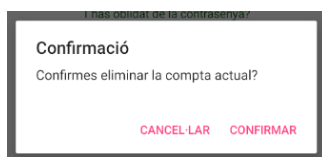
Interaccions disponibles a l'inici de sessió:

- Quadre de text pel "nick" (1): és on es pot introduir el "nick" de l'usuari tal com ja s'indica.
- Quadre de text per la contrasenya (2): és on es pot introduir la contrasenya de l'usuari tal com s'indica en la mateixa pantalla.
- Text clicable d'oblit de contrasenya (3): funcionalitat per si un usuari existent ja té un compte creat però s'ha oblidat de la seva contrasenya. La funcionalitat de recuperació de contrasenya s'explica en l'apartat del [disseny de la seguretat](#).
- Botó per tancar sessió (4): desplega un quadre de diàleg per confirmar que es vol sortir del compte. Si es confirma és tencra la sessió de l'usuari actual i s'obre el menú principal.



Il·lustració 92: Quadre de diàleg per confirmar el tancament de sessió

- Botó d'inici de sessió (5): permet iniciar sessió i si s'introdueix correctament la informació de l'usuari obre el menú principal amb totes les funcions disponibles. Si s'introdueix la contrasenya malament o no s'introdueix cap dada en algun dels quadres de text apareixen avisos personalitzats depenen del problema.
- Text clicable de registre d'un usuari (6): text que si es prem, obre la pantalla de registre d'usuaris.
- Text clicable de donada de baixa (7): text que si es prem, obre un quadre de diàleg per confirmar si es vol eliminar el compte actual.

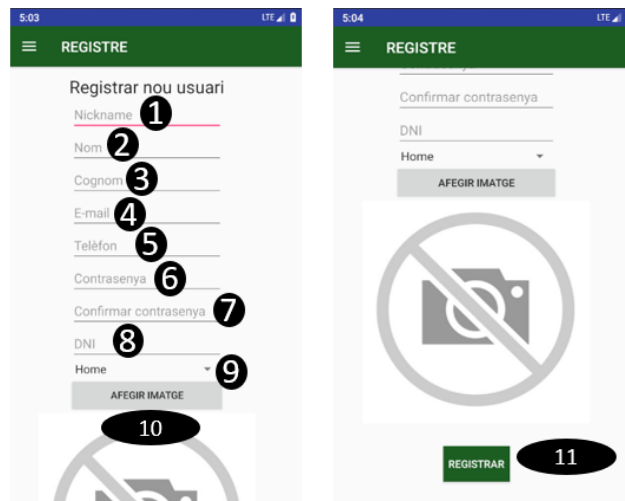


Il·lustració 93: Quadre de diàleg per confirmar l'eliminació del compte

Cal destacar que per poder eliminar un usuari primer s'ha d'haver iniciat sessió amb aquest. Per aquesta raó, si es clica l'opció de donar de baixa sense haver iniciat sessió, l'aplicació també ho especifica amb un missatge.

3. Registre d'usuari

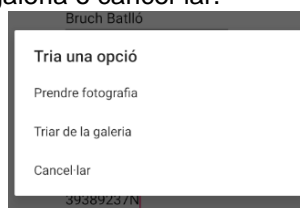
Aquesta és la pantalla encarregada d'agafar tota la informació necessària per poder registrar un usuari a l'aplicació. Hi ha set entrades disponibles per entrar informació, un selector i el botó per registrar tota la informació entrada. Els set espais disponibles són els que permeten introduir la informació del "nick", nom, cognom, correu electrònic, telèfon, contrasenya i confirmació de la contrasenya. El selector és per triar el sexe de l'usuari i el botó per registrar la informació entrada.



Il·lustració 94: Pantalla de registre buida

Interaccions disponibles al registre:

- Quadre de text pel “nick” (1): és on l'usuari ha d'introduir el seu “nick”.
- Quadre de text pel nom (2): és on l'usuari ha d'introduir el seu nom.
- Quadre de text pel cognom (3): és on l'usuari ha d'introduir el seu cognom.
- Quadre de text pel correu electrònic (4): és on l'usuari ha d'introduir el seu correu electrònic.
- Quadre de text pel telèfon (5): és on l'usuari ha d'introduir el telèfon i en aquest cas al quadre de text només s'hi podran introduir números.
- Quadre de text per la contrassenya (6): és on l'usuari ha d'introduir la contrassenya.
- Quadre de text per confirmar la contrassenya (7): és on l'usuari ha d'introduir la mateixa contrassenya que l'anterior per confirmar-la.
- Quadre de text pel DNI (8): és on l'usuari ha d'introduir el seu DNI. En aquest cas es comprova que el DNI sigui correcte.
- *Spinner* per triar el gènere (9): és un *spinner* on es pot triar entre home o dona depenen del gènere de l'usuari.
- Botó per afegir imatge (10): obre un quadre d'opcions amb l'opció de prendre una fotografia, triar una foto de la galeria o cancel·lar.



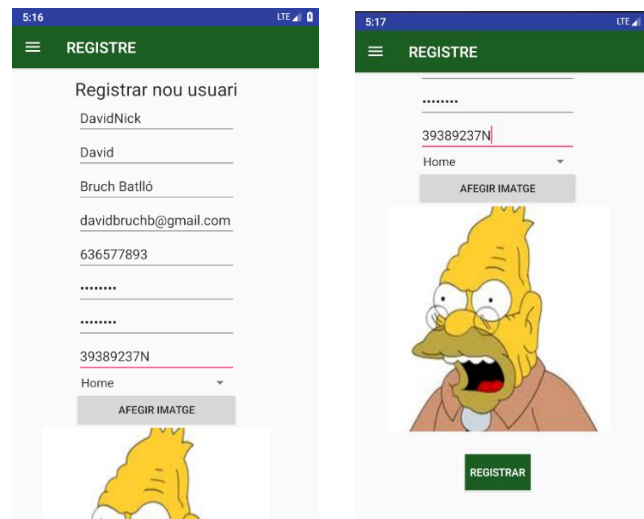
Il·lustració 95: Quadre d'opcions per importar la imatge d'usuari

- Botó per registrar (11): en clicar-se comprova que totes les dades entrades siguin correctes i que no estiguin buides. Si les dades introduïdes compleixen les especificacions, el registre es completa correctament, si no poden aparèixer diferents avisos d'errors, depenen del problema.

En aquest cas com que s'han d'entrar moltes dades s'han tingut en compte diferents errors o excepcions que poden fer aparèixer un usuari a l'hora d'introduir les dades:

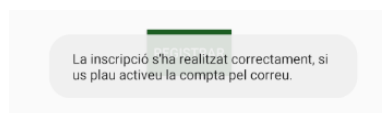
- a) En haver-hi tants requadres per entrar dades un error molt possible és que l'usuari es deixi algun requadre buit i intenti registrar igualment. En aquests casos apareixerà un missatge d'error dient exactament el camp que s'ha deixat buit si només se n'ha deixat

- un i el camp més alt en la llista si se n'ha deixat més d'un buit. A més a més, si deixa totes les entrades buides, també sortirà un missatge d'error personalitzat per aquest cas.
- b) Un cop s'han omplert totes les entrades d'informació es comprova que la contrasenya i la seva confirmació siguin iguals. Si no són iguals, apareix un missatge d'error especificant-ho.
 - c) Hi ha 4 possibles excepcions relacionades amb la base de dades a l'hora de registrar un nou usuari. La primera és que no es pot registrar un usuari amb un "nick" ja utilitzat per un altre usuari existent, ja que el "nick" és la clau primària de la base de dades. Les altres tres són que no es poden repetir el correu electrònic, el telèfon o el DNI amb cap altre usuari per raons de seguretat en la identitat.



Il·lustració 96: Pantalla de registre omplerta

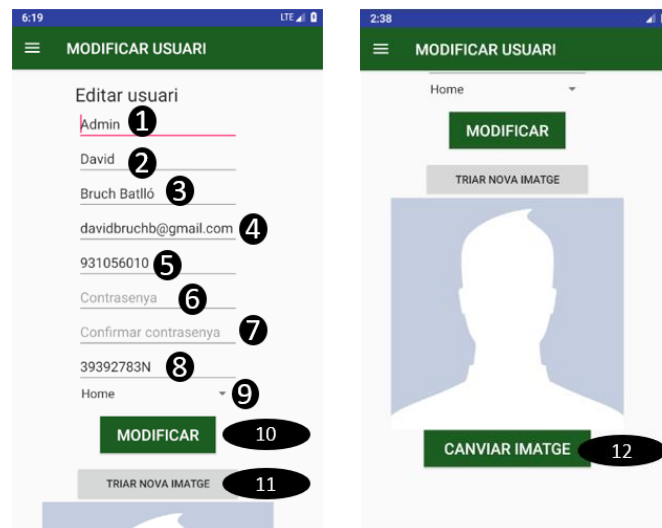
Quan un usuari aconsegueix introduir totes les dades correctament sense que aparegui cap dels errors o excepcions explicats anteriorment es guarden les dades d'aquest usuari a la base de dades i apareix un missatge que especifica que per utilitzar el compte primer has d'activar-lo a través del correu electrònic que has introduït. Aquesta activació s'explica millor a l'apartat de [disseny de la seguretat](#).



Il·lustració 97: Missatge de confirmació del registre

4. Editar usuari

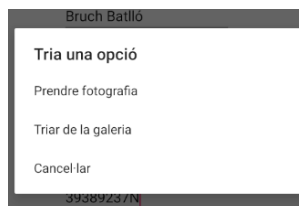
En clicar l'opció d'editar usuari del menú lateral desplegable apareix una pantalla molt semblant a la de registre amb 7 entrades de text completades amb la informació actual de l'usuari, 1 selector pel sexe i un botó per aplicar els canvis. Es veu tota la informació menys la contrasenya, és a dir, sí que la pot canviar si vol igual que qualsevol de les altres dades però apareix buida per més seguretat.



II·lustració 98: Pantalla per editar la informació d'un usuari

Interaccions disponibles a l'editar:

- Quadre de text pel "nick" (1): quadre de text modificable que mostra el "nick" actual de l'usuari.
- Quadre de text pel nom (2): quadre de text modificable que mostra el nom actual de l'usuari.
- Quadre de text pel cognom (3): quadre de text modificable que mostra el cognom actual de l'usuari.
- Quadre de text pel correu electrònic (4): quadre de text modificable que mostra el correu electrònic actual de l'usuari.
- Quadre de text pel telèfon (5): quadre de text modificable que mostra el telèfon actual de l'usuari. A més a més, només s'hi podran introduir números.
- Quadre de text per la contrasenya (6): quadre de text que no mostra la contrasenya actual per seguretat però que permet modificar-la.
- Quadre de text per confirmar la contrasenya (7): quadre de text on l'usuari ha de confirmar la nova contrasenya si la vol modificar.
- Quadre de text pel DNI (8): quadre de text modificable que mostra el DNI actual de l'usuari.
- *Spinner* per triar el gènere (9): *spinner* que mostra el gènere sexual de l'usuari actual i que és modificable.
- Botó per modificar les dades canviades (10): en clicar-se comprova que totes les dades siguin correctes i que no estiguin buides (exceptuant la contrasenya per seguretat). Si les dades compleixen les especificacions, la modificació es completa correctament, si no, poden aparèixer diferents avisos d'errors, depenen del problema.
- Botó per triar nova foto (11): es mostra la imatge d'usuari actual i si es vol modificar, en clicar el botó s'obre un quadre d'opcions amb l'opció de prendre una fotografia, triar una foto de la galeria o cancel·lar.



II·lustració 99: Quadre d'opcions per importar la imatge d'usuari

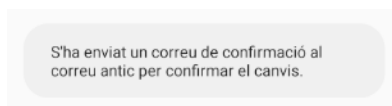
- Botó per canviar imatge (12): si la imatge s'ha modificat correctament amb la funcionalitat anterior al clicar aquest es canviarà la imatge d'usuari per la nova. S'hi s'intenta clicar sense canviar la imatge, apareix un avís especificant-ho.

La diferència principal amb el registre és que aquí en ser una modificació de dades, les dades que no vulguis modificar les deixes iguals a com les has trobat en obrir la pantalla, és a dir, amb la informació actual de l'usuari i amb la contrasenya buida per seguretat.

Igualment, com que si l'usuari vol modificar moltes dades, ha de modificar moltes entrades d'informació, s'han tingut en compte diferents errors o excepcions que poden fer aparèixer els usuaris a l'hora de modificar les dades:

- a) En haver-hi tants requadres per modificar dades un error molt possible és que l'usuari deixi algun requadre buit i intenti modificar igualment. En aquests casos apareixerà un missatge d'error mostrant exactament el camp que s'ha deixat buit si només se n'ha deixat un i el camp més alt en la llista si se n'ha deixat més d'un. A més a més, si deixa totes les entrades buides, també sortirà un missatge d'error personalitzat per aquest cas.
- b) Un cop s'han omplert totes les entrades d'informació menys la contrasenya i la seva confirmació a no ser que l'usuari vulgui canviar la contrasenya. Si l'usuari vol canviar la contrasenya, es comprova si la contrasenya nova i la seva confirmació són iguals, si no són iguals, apareix un missatge d'error especificant-ho.
- c) Hi ha 4 possibles excepcions relacionades amb la base de dades, que són les mateixes que amb el registre.

Quan un usuari aconsegueix modificar les dades que vol correctament sense que aparegui cap dels errors nomenats anteriorment s'envia un correu al correu antic si s'ha modificat aquest o al correu actual sinó per confirmar els canvis fets. Si no es confirmen els canvis fets a través del correu, no es duren a terme. A més a més quan es produeixi una modificació i es confirma amb el botó, el sistema ens treu de la sessió actual per haver de tornar a entrar, ja que l'usuari pot haver modificat el "nick" o la contrasenya, que són les entrades necessàries per iniciar sessió.



Il·lustració 100: Missatge del fet que s'ha enviat un correu de confirmació

Confirmació dels canvis fets en la compta del club de tennis i pàdel



Club Tennis Navàs <clubtennisn@gmail.com>

19:00

Para: davidbruchb@gmail.com

Hola Admin,

Per confirmar els canvis fets en la informació de la teva compta clica el següent enllaç.

[Enllaç de confirmació de la modificació de compta.](#)

Il·lustració 101: Correu electrònic de confirmació

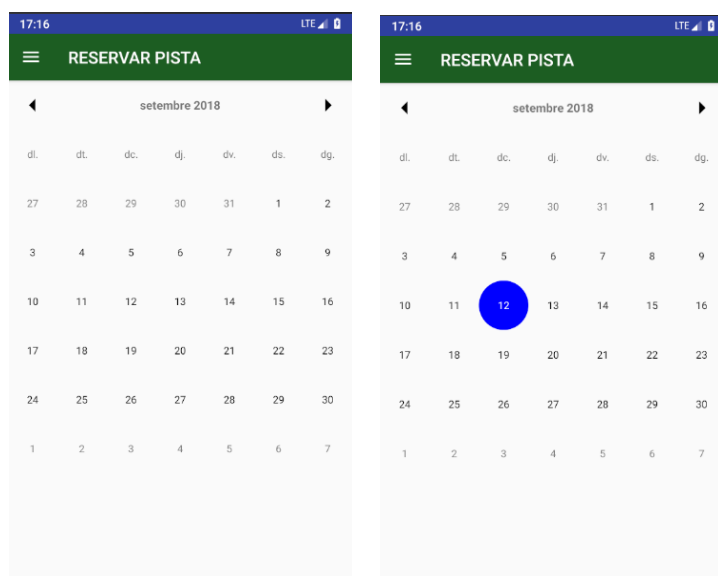
5. Reservar pista

La reserva d'una pista està formada per dues pantalles, una que és un calendari per triar el dia de la reserva i l'altre per poder triar la pista i les hores d'aquesta reserva.

5.1. Calendari de reserva

En clicar l'opció de reservar pista sigui pel menú principal o pel menú desplegable lateral, la primera pantalla que ens trobem és la d'un calendari, el qual mostra el més actual i ens dona la possibilitat de canviar de mes utilitzant les fletxes negres de dalt o simplement arrossegant la

pantalla cap a una banda amb el dit. Si intentem clicar un dia anterior al dia actual, apareix un error i ens deixa tornar a clicar un altre dia fins que en cliquem un de disponible.



Il·lustració 102: Calendari disponible per triar el dia de la reserva

Quan un usuari clica una data correctament, és a dir, clica una data posterior al dia actual o el mateix dia actual el sistema ens redirigeix cap a una pantalla on es pot triar la pista i les hores del dia seleccionat. Per altra banda, si se selecciona una data anterior al dia actual o una data molt llunyana (més de 3 mesos) apareixen avisos personalitzats.

5.2. Pista i hores d'un dia concret

Aquesta pantalla és una de les més complexes de tot el projecte, ja que interacciona amb molts elements diferents i és capaç de dur a terme moltes funcionalitats. Primer de tot podem observar que hi ha 4 botons grans de color verd que simbolitzen les 4 pistes del club. A sota hi ha tot un seguit d'hores que inicialment quan s'entra a la pantalla estan desactivades i no es poden clicar. Finalment un botó verd per guardar una reserva, i un altre per eliminar les reserves que hagi pogut fer aquell usuari en aquell dia i pista.



Il·lustració 103: Pantalla de reserva

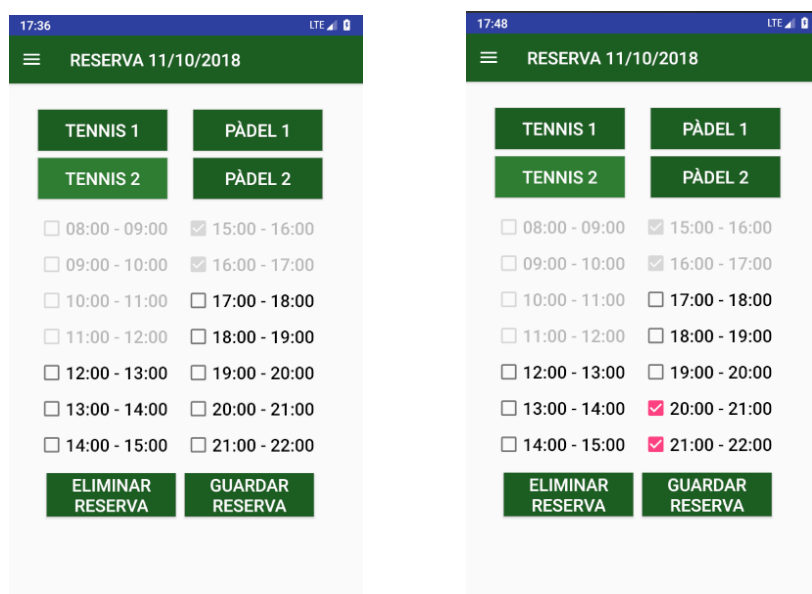
Interaccions disponibles en reservar:

- Botons de pista (1): 4 botons dels quals al clicar-ne un es carregen les hores d'aquella pista i canvia de color per deixar a l'usuari modificar els *radiobuttons*.
- *Radiobuttons* per seleccionar les hores (2): *radiobuttons* que permeten seleccionar les hores que estiguin lliures en la pista seleccionada anteriorment. Té un límit de com a

màxim 4 hores per un mateix usuari en una pista i s'informa a l'usuari s'hi s'intenta excedir.

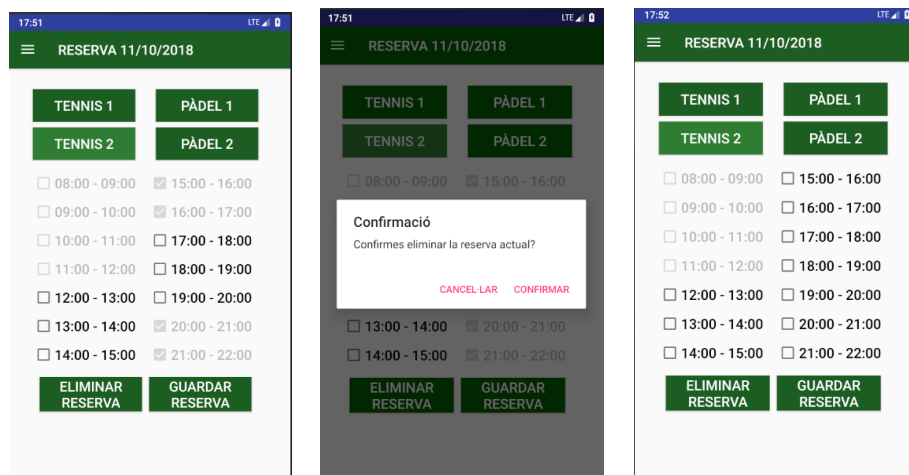
- Botó per guardar reserva (3): botó que permet guardar les hores de la reserva que s'hagin clicat anteriorment. Com ja s'ha explicat anteriorment hi ha un límit de 4 hores i a més a més, si es clica el botó sense haver reservat res, també es produeix un avís.
- Botó per eliminar reserva (4): botó que permet eliminar les hores reservades de la pista actual. Si l'usuari no té cap reserva a la pista actual, apareix un missatge especificant-ho.

Primer de tot l'usuari ha de clicar una de les pistes per poder habilitar les hores d'aquell dia. Si hi ha hores ocupades per algú altre, es quedaran desactivades per mostrar a l'usuari actual que estan ocupades i per evitar que les pugui clicar. A més a més, si alguna de les hores ocupades és seva, sortirà desactivada però marcada també per mostrar a l'usuari que està ocupada però per ell mateix. A continuació l'usuari podrà clicar les hores que vulgui reservar d'aquell dia i d'aquella pista que estiguin habilitades per poder-les ocupar ell. Un cop marcades només haurà de clicar el botó de guardar reserva i es guardarà la reserva actual.



Il·lustració 104: Pantalles de reserva amb hores ocupades per un altre usuari, per l'usuari actual i hores clicades per reservar-ne de noves.


En aquesta pantalla també hi ha disponible eliminar les reserves fetes independentment de les hores ocupades en una pista concreta. Quan es clica apareix un requadre per confirmar l'eliminació de les reserves que aquest usuari pugui tenir aquell dia en la pista seleccionada. Si es cancel·la, no passa res i si s'accepta, apareix un missatge per confirmar l'eliminació i desapareixen visualment les hores reservades en aquella pista.



Il·lustració 105: Procés d'eliminació de les reserves d'una pista

Si l'usuari du a terme una reserva correcta, apareix un missatge per confirmar que la reserva s'ha guardat correctament i automàticament arriba una notificació al correu de l'usuari que ha fet la reserva per recordar-li el dia, la pista i les hores que ha reservat.

Recordatori de reserva.

 Club Tennis Navàs <clubtennisn@gmail.com>
19:55

Para: davidbruchb@gmail.com

Hola Admin,
La teva reserva ha sigut guardada pel dia 2018-10-11 a la pista de tennis 2 durant les hores:
15:00-16:00
16:00-17:00

Il·lustració 106: Correu electrònic del recordatori de reserva.

Finalment s'ha afegit en el disseny de les reserves elements de seguretat per evitar que un sol usuari pugui per exemple reservar totes les hores d'un dia. Primer de tot s'evita que un usuari pugui reservar una pista d'un dia anterior a l'actual. A més a més, no es pot reservar una pista per fora d'un interval de 3 mesos de temps des del dia actual, ja que per un club de tennis i pàdel no és necessari poder reservar amb tanta antelació.

Per evitar el problema del fet que un usuari pugui reservar totes les hores que vulgui s'ha limitat de manera que un sol usuari no pot reservar més de 4 hores en una mateixa pista i a més a més, no pot reservar les mateixes hores entre dues pistes diferents un mateix dia.

6. Reserves actives

Aquest botó del menú lateral porta cap a una pantalla on es pot veure la informació de les reserves actives (reserves per partits que encara s'han de jugar) separades per dies i dins dels dies per pistes. D'aquesta manera un usuari pot veure ràpidament amb una ullada quines reserves actives té en aquell moment sense necessitat d'haver d'anar entrant als diferents dies del calendari de reserves.



Il·lustració 107: Pantalla de reserves actives

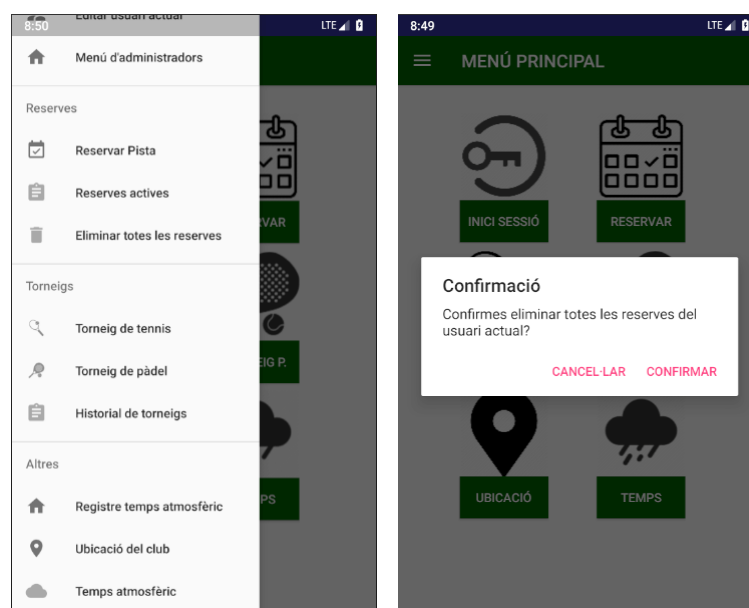
Si un usuari no té cap reserva activa i clica el botó li apareix un missatge avisant que no té cap reserva activa i el porta cap al menú principal.



Il·lustració 108: Missatge de cap reserva activa que porta al menú principal

7. Elimina totes les reserves

És una opció extra de les reserves i una extensió de l'eliminació de les reserves d'una pista mencionada anteriorment, ja que aquest botó, que només apareix al menú desplegable lateral, permet eliminar totes les hores reservades de l'usuari actual independentment del dia i de la pista al mateix temps. Com que és un canvi important en les reserves d'un usuari, quan es clica primer apareix un requadre per confirmar o no l'eliminació de totes les reserves actuals. Si es cancel·la, no passa res i si es confirma, apareix un missatge que et diu que s'han eliminat totes les hores reservades correctament.

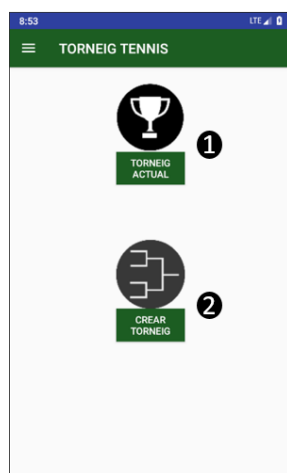


Il·lustració 109: Botó per eliminar i diàleg de confirmació

Si un usuari clica el botó d'eliminar totes les reserves sense tenir cap reserva activa en aquell moment, li apareix un missatge personalitzat del fet que no té cap reserva activa.

8. Torneig de tennis

El torneig de tennis està format per diferents pantalles que duen a terme totes les diferents opcions d'un torneig. La primera pantalla que ens trobem en clicar el botó del torneig de tennis és una amb dues opcions diferents.



Il·lustració 110: Pantalla d'inici a l'entrar al torneig de tennis

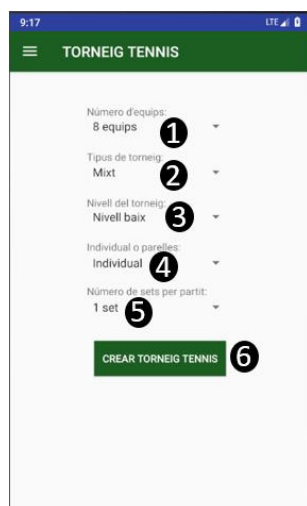
Interaccions disponibles en reservar:

- Botó de torneig actual (1): en clicar-se permet entrar al torneig actual de tennis, sempre que n'hi hagi un de creat.
- Botó de creació de torneig (2): només hi poden accedir els administradors i permet entrar a una pantalla per crear un nou torneig de tennis.

L'opció de sobre és la que es clica per entrar a veure el torneig de tennis quan n'hi ha un de creat. Quan no n'hi ha cap de creat simplement surt un missatge dient-ho i no ens deixa entrar. Per aquesta raó també hi ha el segon botó que porta a una pantalla per crear el torneig temporal però que només hi poden accedir els administradors de l'aplicació, és a dir, els usuaris que es troben dins de la taula d'administradors a la base de dades tal com s'explica al [disseny de la base de dades](#). Si un usuari normal intenta entrar a la pantalla de creació de torneig, li apareix un missatge recordant-li que no està autoritzat per entrar allà.

8.1. Crear torneig de tennis

En el cas del torneig de tennis el primer que trobem en aquesta pantalla són 5 selectors diferents per triar les diferents característiques del torneig de tennis i un botó per crear-lo.



Il·lustració 111: Pantalla per la creació del torneig de tennis

Interaccions disponibles al crear el torneig de tennis:

- *Spinner* del nombre d'equips (1): és el nombre d'equips que formaran el torneig actual i pot ser de tres tipus. Les opcions disponibles són la de 8 equips, 16 equips i 32 equips depenent de si l'administrador vol fer un torneig més petit o més gran.
- *Spinner* del tipus de torneig (2): el tipus de torneig és el sexe que podran tenir els integrants dels equips i hi ha 3 opcions disponibles. El tipus de torneig pot ser mixt (hi poden haver tant dones com homes), pot ser femení (*només* dones) o pot ser masculí (*només* homes).
- *Spinner* del nivell del torneig (3): és una opció que servirà per guiar una mica el nivell que es vol dels jugadors en aquest torneig de tennis. Té 4 opcions disponibles que són nivell baix, nivell mitjà, nivell alt o tots els nivells quan és indiferent.
- *Spinner* dels equips individuals o per parelles (4): tal com el seu nom indica aquest selector serveix per poder triar si els equips seran individuals (1 jugador) o per parelles (2 jugadors).
- *Spinner* del nombre de sets per partit (5): és el nombre de sets necessaris per guanyar els partits del torneig de tennis que es vol crear i està format per 3 opcions. Les opcions són 1 set, 3 sets o 5 sets depenent de la llargada dels partits.
- Botó de creació de torneig (6): en clicar-se es crea un torneig de tennis amb les especificacions triades anteriorment. Si ja hi ha un torneig de tennis creat no es deixarà crear un altre fins que l'actual acabi.

Un cop l'usuari ha triat les opcions que vol pel torneig només ha de clicar el botó per crear-lo, el qual farà aparèixer un missatge que confirma que s'ha creat correctament. Si s'intenta clicar el botó quan ja hi ha un torneig creat apareix un missatge personalitzat avisant que ja hi ha un torneig existent.

8.2. Torneig actual de tennis

Aquesta pantalla té 4 opcions diferents i la part posterior informa sobre les característiques del torneig de tennis actual, que són les 5 característiques que un administrador ha donat prèviament a l'hora de crear el torneig de tennis actual. En aquesta pantalla l'usuari té a l'abast 4 opcions ben diferenciades que són el registre al torneig de tennis, la llista d'equips registrats, l'arbre de partits i la taula de partits.



Il·lustració 112: Pantalla del torneig de tennis

Interaccions disponibles al torneig de tennis:

- Botó de registre al torneig de tennis (1): botó que permet registrar-se al torneig mentre aquest no estigui encara ple.
- Botó dels jugadors del torneig de tennis (2): botó que obre una pantalla on es pot observar la llista dels participants al torneig.
- Botó de l'arbre de partits de tennis (3): funcionalitat que obre l'arbre de partits del torneig actual de tennis sempre que ja estigui actiu. Si no s'avisava que el torneig encara no està actiu perquè encara s'està omplint.
- Botó de la taula de partits de tennis (4): funcionalitat que obre una pantalla amb la informació dels partits i que permet actualitzar-los pels jugadors que hi participen.

8.2.1. Registre al torneig de tennis

Aquest botó de la pantalla permet a l'usuari registrar-se al torneig de tennis actual. Aquest registre canvia una mica depenent de si el torneig és individual o és per parelles.

a) Registre en un torneig individual

En aquest cas com que els equips seran d'una sola persona al clicar el botó apareix un requadre que li demana a l'usuari que introdueixi un nom del seu equip. Si l'usuari cancel·la la introducció del nom al requadre, no passa res, en canvi, si introdueix un nom ja utilitzat per un altre equip o no introdueix cap nom i confirma apareixerà un missatge d'error personalitzat depenent del problema.

Il·lustració 113: Requadre per introduir el nom d'equip

Si l'usuari que intenta registrar-se al torneig és d'un gènere que no és igual al tipus de torneig també surt un missatge especificant-ho. A més a més, si un jugador prova de tornar a registrar-se quan ja està registrat en un torneig el sistema no el deixarà i apareixerà un missatge dient que el jugador actual ja participa en el torneig actual. Finalment si el torneig està ja ple i un usuari prova de registrar-s'hi li apareix un missatge especificant que el torneig ja està ple.

Quan l'usuari introdueix un nom d'equip disponible i confirma el registre quedarà registrat dins del torneig actual amb el nom d'equip que ha introduït.

Equip registrat correctament al torneig

Il·lustració 114: Equip registrat correctament

b) Registre en un torneig per parelles

En aquest cas com que els equips són de 2 persones en clicar el registre el primer que surt és un quadre per introduir el nom del segon integrant.

Tant pel primer integrant que és l'usuari actual que està clicant com pel segon integrant es comprova que es compleixi que el gènere sigui el que toca per aquell torneig i que el nom del segon integrant existeixi dins dels usuaris actuals, i si no existeix, surti un missatge d'error. També es comprova tant pel segon integrant com pel primer que no estiguin ja en un altre equip d'aquest mateix torneig, i si és així apareix un missatge d'error especificant-ho. A més a més, si es prova d'entrar un nom d'usuari buit, també apareix un missatge d'error per avisar-ho.

Quan l'usuari aconsegueix introduir el nom del segon integrant del seu equip correctament apareix un altre requadre que permet introduir el nom de l'equip dels dos jugadors. En aquest cas les excepcions i errors possibles a l'entrar el nom de l'equip són igual als que poden aparèixer quan s'introdueix el nom de l'equip individual.

Finalment si el torneig està ja ple i un usuari prova de registrar-s'hi li apareix un missatge especificant que el torneig ja està ple. Quan l'usuari introdueix un nom d'usuari ("nick") disponible per ser el seu company, un nom d'equip disponible i confirma el registre quedarà registrat dins del torneig actual amb el nom d'equip que ha introduït juntament amb el seu company.

8.2.2. Llista d'equips del torneig de tennis

En aquesta pantalla hi podem trobar una llista de tots els equips que s'han registrat al torneig actual i la seva informació, és a dir, els integrants de cada equip, el seu "nick", el seu nom, el seu correu electrònic, la seva imatge de perfil i el seu telèfon per si s'ha de contactar amb ells. Aquesta pantalla és útil tant pels jugadors com pels administradors, ja que permet veure ràpidament els integrants de cada equip i les seves dades. A més a més, aquesta pantalla també té 2 botons a la part superior que serveixen per iniciar i per finalitzar el torneig i que *només* poden clicar els administradors, ja que si un usuari normal prova de clicar-los, li apareix un missatge d'error que diu que no està autoritzat.



Il·lustració 115: Llista d'equips al torneig de tennis

Interaccions disponibles al torneig de tennis:

- Botó per iniciar torneig (1): botó només disponible per administradors que permet iniciar un torneig un cop està ple.
- Botó per finalitzar torneig (2): botó només disponible per administradors que permet finalitzar un torneig en qualsevol moment.

Si l'usuari que clica el botó d'inici de torneig és un administrador el deixarà però *només* s'activarà el torneig si el registre a aquest està ple, si no està ple l'avisarà un missatge que encara no es pot activar el torneig per començar els partits. En canvi, si el torneig ja està ple i l'administrador clica el botó d'inici s'activarà el torneig i es generaran la primera ronda de partits del torneig.

De la mateixa manera el botó de finalitzar torneig només el pot clicar un administrador. El torneig es pot eliminar quan l'administrador vulgui estigui el torneig acabat o no.

8.2.3. Arbre de partits de tennis

En aquesta pantalla només si pot entrar quan el torneig ha estat activat per un administrador, ja que com que mostra la progressió dels partits en forma d'arbre només es podran mostrar quan s'activi el torneig i comencin els partits. Si es prova d'entrar a aquesta pantalla clicant el botó de l'arbre de partits sense que el torneig estigui actiu apareix un missatge avisant que el torneig encara no està actiu i no deixant entrar a la pantalla.

Si el torneig ja està actiu, pot entrar qualsevol usuari a veure l'arbre de partits en qualsevol moment fins que el torneig finalitzi, ja que l'arbre va canviant a mesura que el torneig avança mostrant els partits entre equips i els seus resultats.



Il·lustració 116: Arbre de partits del torneig de tennis en una etapa inicial



Il·lustració 117: Arbre de partits del torneig de tennis en una etapa mitjana i final

8.2.4. Taula de partits de tennis

Igual que en la pantalla d'arbre de partits aquesta pantalla només es pot visualitzar quan el torneig ha estat activat per un administrador i hi poden entrar tots els usuaris per veure com avancen els partits estiguin o no participant en el torneig.

En aquesta pantalla es poden veure els partits que s'estan fent i a més a més és a on els jugadors actualitzen els partits introduint el resultat del partit i posant si han guanyat o perdut utilitzant els botons de victòria i derrota.

E1	E2	Resultat	Guanyador	Perdedor
5555	4444	6-4	4444	5555
7777	1	7-5	1	7777
6666	8888	6-1	8888	6666
333333	222222	6-0	333333	222222
4444	1	6-3	1	4444
6666	333333	6-2	6666	333333
1	6666	6-4	6666	1

Il·lustració 118: Taula de partits del club de tennis

Interaccions disponibles a la tala de partits de tennis:

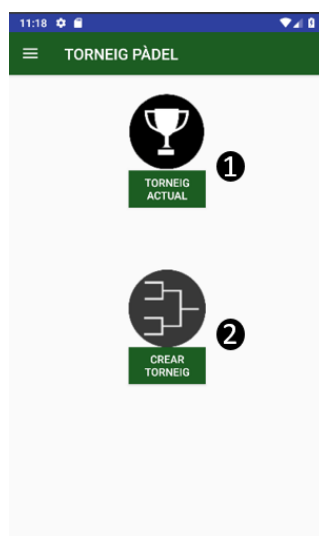
- Quadre de text per introduir el marcador (1): quadre on s'introdueix el marcador del partit abans de registrar el guanyador o el perdedor. Si ja s'ha registrat anteriorment, s'actualitza automàticament i és modificable igualment.
- Botó per registrar victòria (2): botó que permet registrar la victòria d'un usuari amb el marcador introduït anteriorment.
- Botó per registrar derrota (3): botó que permet registrar la derrota d'un usuari amb el marcador entrat anteriorment.

Si un jugador que no té cap partit actiu, ja sigui perquè no participa en el torneig o perquè ja han eliminat el seu equip, intenta entrar un marcador de victòria o derrota li sortirà un missatge d'error especificant que no té cap partit actiu. Si un jugador intenta posar una victòria quan ja està registrada en el partit o una derrota quan ja està registrada li sortirà també un missatge d'error especificant-ho. I si prova de clicar el botó de victòria o derrota sense haver posat un marcador abans apareixerà un missatge d'error personalitzat dient que primer s'ha d'introduir un marcador.

El funcionament de les rondes de partits i de la seva estructura s'explica amb més detall en la [implementació](#).

9. Torneig de pàdel

El torneig de pàdel està format per diferents pantalles que duen a terme totes les diferents opcions. La primera pantalla que ens trobem en clicar el botó del torneig de pàdel és una amb dues opcions diferents.



Il·lustració 119: Pantalla d'inici a l'entrar al torneig de pàdel

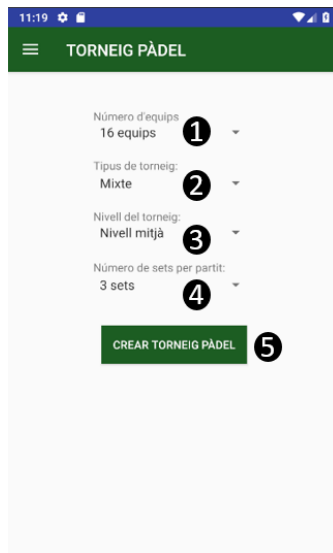
Interaccions disponibles a l'inici del torneig de pàdel:

- Botó de torneig actual (1): en clicar-se permet entrar al torneig actual de pàdel, sempre que n'hi hagi un de creat.
- Botó de creació de torneig (2): només hi poden accedir els administradors i permet entrar a una pantalla per crear un nou torneig de pàdel.

L'opció de sobre és la que es clica per entrar a veure el torneig de pàdel quan n'hi ha un de creat, en canvi, quan no n'hi ha cap de creat simplement surt un missatge dient-ho i no ens deixa entrar. Per aquesta raó també hi ha el segon botó que porta a una pantalla per crear el torneig temporal de pàdel però que només hi poden accedir els administradors de l'aplicació. Si un usuari normal intenta entrar a la pantalla de creació de torneig, li apareix un missatge recordant-li que no està autoritzat per entrar-hi.

9.1. Crear torneig de pàdel

En el cas de la creació del torneig de pàdel el primer que trobem en aquesta pantalla són 4 selectors diferents per triar les diferents característiques del torneig de pàdel i un botó per crear-lo.



Il·lustració 120: Pantalla per la creació d'un torneig de pàdel

Interaccions disponibles en crear el torneig de pàdel:

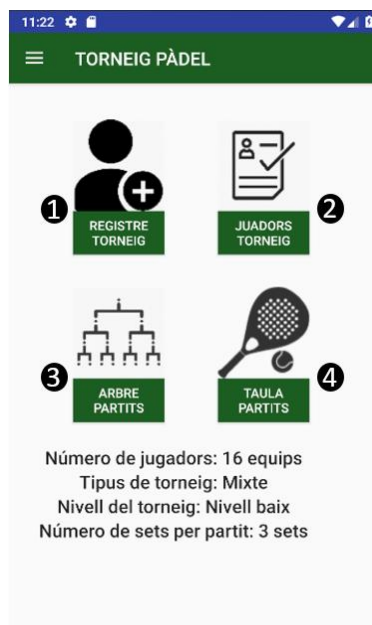
- *Spinner* del nombre d'equips (1): és el nombre d'equips que formaran el torneig actual i pot ser de tres tipus. Les opcions disponibles són la de 8 equips, 16 equips i 32 equips depenent de si l'administrador vol fer un torneig més petit o més gran.
- *Spinner* del tipus de torneig (2): el tipus de torneig és el sexe que podran tenir els integrants dels equips i hi ha 3 opcions disponibles. El tipus de torneig pot ser mixt (hi poden haver tant dones com homes), pot ser femení (*només* dones) o pot ser masculí (*només* homes).
- *Spinner* del nivell del torneig (3): és una opció que servirà per guiar una mica el nivell que es vol dels jugadors en aquest torneig de pàdel. Té 4 opcions disponibles que són nivell baix, nivell mitjà, nivell alt o tots els nivells quan és indiferent.
- *Spinner* del nombre de sets per partit (4): és el nombre de sets necessaris per guanyar els partits del torneig de pàdel que es vol crear i està format per 3 opcions. Les opcions són 1 set, 3 sets o 5 sets depenent de la llargada dels partits.
- Botó de creació de torneig (5): al clicar-se es crea un torneig de pàdel amb les especificacions triades anteriorment. Si ja hi ha un torneig de pàdel creat, no es deixarà crear un altre fins que l'actual acabi.

A diferència del torneig de tennis en el torneig de pàdel no es pot triar si els equips són individuals o per parelles, ja que en el pàdel els equips sempre són per parelles.

Un cop l'usuari ha triat les opcions que vol pel torneig només ha de clicar el botó per crear-lo, el qual farà aparèixer un missatge que confirma que s'ha creat correctament. Si s'intenta clicar el botó quan ja hi ha un torneig creat apareix un missatge personalitzat avisant que ja hi ha un torneig existent.

9.2. Torneig actual de pàdel

Aquesta pantalla té 4 opcions diferents i la part posterior informa sobre les característiques del torneig de pàdel actual, que són les 4 característiques que un administrador ha donat anteriorment a l'hora de crear el torneig. En aquesta pantalla l'usuari té a l'abast 4 opcions ben diferenciades que són el registre al torneig de pàdel, la llista d'equips registrats, l'arbre de partits i la taula de partits.

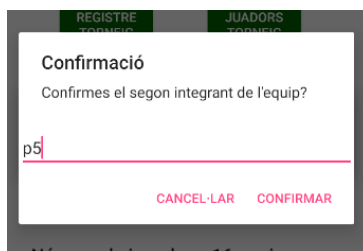


Il·lustració 121: Pantalla del torneig de pàdel

9.2.1. Registre al torneig de pàdel

Aquest botó de la pantalla permet a l'usuari registrar-se al torneig de pàdel actual. A diferència del torneig de tennis en el de pàdel només hi ha equips per parelles pel que el registre serà sempre per parelles.

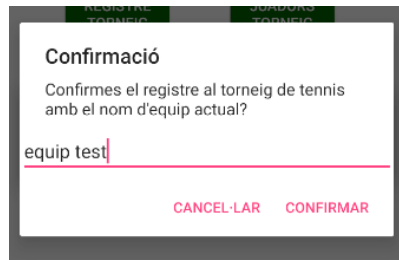
Com que els equips són de 2 persones, en clicar el registre el primer que surt és un quadre per introduir el nom del segon integrant.



Il·lustració 122: Requadre per introduir el segon integrant

Tant pel primer integrant que és l'usuari actual que està clicant com pel segon integrant es comprova que es compleixi que el gènere sigui el que toca per aquell torneig i que el nom del segon integrant existeixi dins dels usuaris actuals i si no existeix, surti un missatge d'error. També es comprova tant pel segon integrant com per primer que no estiguin ja en un altre equip d'aquest mateix torneig i si és així apareix un missatge d'error especificant-ho. A més a més, si es prova d'entrar un nom d'usuari buit, també apareix un missatge d'error per avisar-ho.

Quan l'usuari aconsegueix introduir el nom del segon integrant del seu equip correctament apareix un altre requadre que permet introduir el nom de l'equip dels dos jugadors. Si l'usuari cancel·la la introducció del nom al requadre, no passa res, en canvi, si introdueix un nom ja utilitzat per un altre equip o no introdueix cap nom i confirma apareixerà un missatge d'error personalitzat depenent del problema.



Il·lustració 123: Requadre per introduir el nom d'equip de pàdel

Finalment si el torneig està ja ple i un usuari prova de registrar-s'hi li apareix un missatge especificant que el torneig ja està ple.

Quan l'usuari introdueix un nom d'usuari ("nick") disponible per ser el seu company, un nom d'equip disponible i confirma el registre quedarà registrat dins del torneig actual amb el nom d'equip que ha introduït juntament amb el company que ha introduït.

9.2.2. Llista d'equips del torneig de pàdel

En aquesta pantalla hi podem trobar una llista de tots els equips que s'han registrat al torneig actual i la seva informació, es a dir, els integrants de cada equip, el seu "nick", el seu nom, el seu correu electrònic, la seva imatge de perfil i el seu telèfon per si s'ha de contactar amb ells. Aquesta pantalla és útil tant pels jugadors com pels administradors, ja que permet veure ràpidament els integrants de cada equip i les seves dades. A més a més, aquesta pantalla també té 2 botons a la part superior que serveixen per iniciar i per finalitzar el torneig i que *només* poden clicar els administradors, ja que si un usuari normal prova de clicar-los li apareix un missatge d'error que diu que no està autoritzat.



Il·lustració 124: Llista d'equips al torneig de pàdel

Interaccions disponibles al torneig de pàdel:

- Botó per iniciar torneig (1): botó només disponible per administradors que permet iniciar un torneig un cop està ple.
- Botó per finalitzar torneig (2): botó només disponible per administradors que permet finalitzar un torneig en qualsevol moment.

Si l'usuari que clica el botó d'inici de torneig és un administrador el deixarà però *només* s'activarà el torneig si el registre a aquest està ple, si no està ple el sistema l'avisarà amb un missatge que encara no es pot activar el torneig per començar els partits. En canvi, si el torneig ja està ple i

l'administrador clica el botó d'inici s'activarà el torneig i es generarà la primera ronda de partits del torneig.

De la mateixa manera el botó de finalitzar torneig només el pot clicar un administrador. El torneig es pot eliminar quan l'administrador vulgui, estigui el torneig acabat o no.

9.2.3. Arbre de partits de pàdel

En aquesta pantalla només si pot entrar quan el torneig ha estat activat per un administrador, ja que com que mostra la progressió dels partits en forma d'arbre, només es podrà mostrar quan s'activi el torneig i comencin els partits. Si es prova d'entrar a aquesta pantalla clicant el botó de l'arbre de partits sense que el torneig estigui actiu apareix un missatge avisant que el torneig encara no està actiu i no deixant visualitzar la pantalla.

Si el torneig ja està actiu, pot entrar qualsevol usuari a veure l'arbre de partits en qualsevol moment fins que el torneig finalitzi, ja que l'arbre va canviant a mesura que el torneig avança mostrant els partits entre equips i el seus resultats.



Il·lustració 125: Arbre de partits del torneig de pàdel en una etapa inicial

9.2.4. Taula de partits de pàdel

Igual que en la pantalla d'arbre de partits aquesta pantalla només es pot visualitzar quan el torneig de pàdel ha estat activat per un administrador i hi poden entrar tots els usuaris per veure com avancen els partits estiguin o no participant en el torneig.

En aquesta pantalla es poden veure els partits que s'estan fent i a més a més és on els jugadors actualitzen els resultats d'aquests introduint el resultat del partit i posant si han guanyat o perdut utilitzant els botons de victòria i derrota.



Il·lustració 126: Taula de partits del torneig de pàdel

Interaccions disponibles a la taula de partits de pàdel:

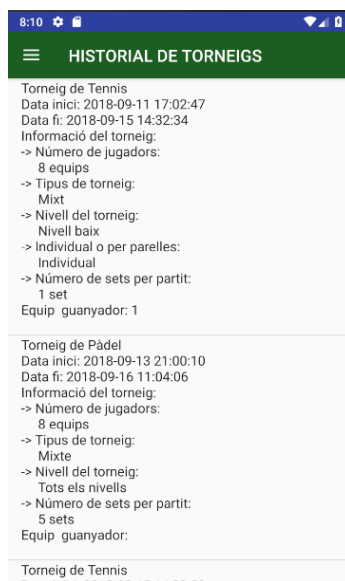
- Quadre de text per introduir el marcador (1): quadre on s'introdueix el marcador del partit abans de registrar el guanyador o el perdedor. Si ja s'ha registrat anteriorment, s'actualitza automàticament i és modificable igualment.
- Botó per registrar victòria (2): botó que permet registrar la victòria d'un usuari amb el marcador introduït anteriorment.
- Botó per registrar derrota (3): botó que permet registrar la derrota d'un usuari amb el marcador entrat anteriorment.

Si un jugador que no té cap partit actiu, ja sigui perquè no participa en el torneig o perquè ja han eliminat el seu equip, intenta entrar un marcador de victòria o derrota li apareixerà un missatge d'error especificant que no té cap partit actiu. Si un jugador intenta posar una victòria quan ja està registrada en el partit o una derrota quan ja està registrada li sortirà també un missatge d'error especificant-ho. I si prova de clicar el botó de victòria o derrota sense haver posat un marcador abans apareixerà un missatge d'error personalitzat dient que primer s'ha d'introduir un marcador.

El funcionament de les rondes de partits i de la seva estructura s'explicarà en més detall en la [implementació](#).

10. Historial de torneigs

El botó de l'historial de torneigs es troba en el menú lateral desplegable i hi pot accedir qualsevol usuari sempre que hagi iniciat sessió abans. Un cop un usuari hi entra pot veure una llista dels diferents torneigs que s'han dut a terme on s'especifica de cada un si era un torneig de tennis o de pàdel, la data d'inici, la data de finalització i l'equip guanyador d'aquella edició del torneig.



Il·lustració 127: Pantalla de l'història de torneigs

Si no apareix guanyador al torneig registrat és que aquella edició es va finalitzar abans d'hora per un administrador.

11. Registre del temps atmosfèric

És una opció que retorna una llista del temps que fa a la pista cada hora i que ve donada per un programa dissenyat amb “python” que utilitzant una llibreria de “OpenWeatherMap”.

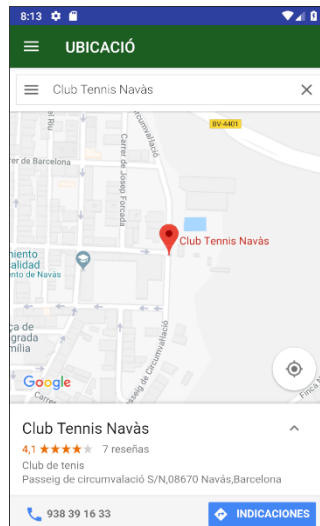
Quan l'usuari entra en aquesta opció es carrega la informació del temps que s'ha anat registrant a la taula perquè pugui veure la variació al llarg del temps.



Il·lustració 128: Pantalla del registre atmosfèric

12. Ubicació del club de tennis i pàdel

El botó de la ubicació es pot trobar tant en el menú principal com en el menú lateral desplegable i hi pot accedir qualsevol usuari sempre que hagi iniciat sessió abans. En clicar aquest botó l'usuari pot visualitzar la ubicació del club gràcies a una crida al “Google Maps” per veure exactament on es troba localitzat per així poder arribar-hi sense problemes.



Il·lustració 129: Pantalla d'ubicació del club

13. Temps atmosfèric al club

El botó del temps atmosfèric es pot trobar tant en el menú principal com en el menú lateral desplegable i hi pot accedir qualsevol usuari sempre que hagi iniciat sessió abans. En clicar aquest botó l'usuari pot visualitzar el temps del dia actual i la predicció dels següents dies en la localització del club gràcies a una crida a "Meteocat" per veure amb exactitud si es podrà jugar a les pistes. Aquesta funció és útil tenint en compte que en ser pistes exteriors si plou, no es pot jugar en elles.



Il·lustració 130: Pantalla del temps atmosfèric al club

14. Contactar amb el club

Aquest botó es troba al final del menú desplegable lateral i mostra el logotip, la direcció, el telèfon i el correu electrònic del club per tal de permetre als usuaris contactar directament si hi ha alguna incidència.

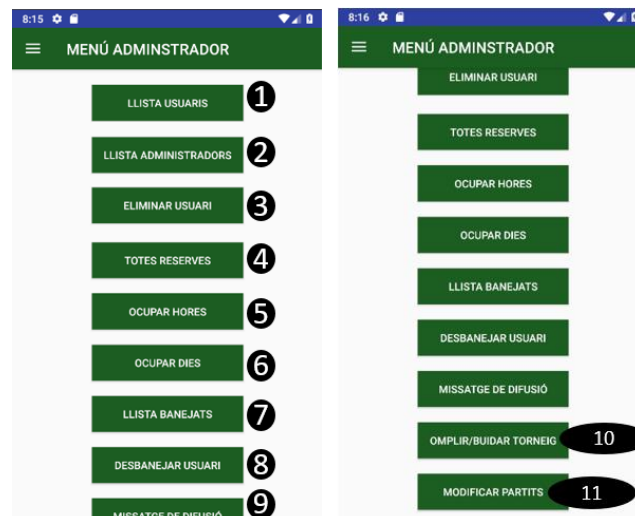


Il·lustració 131: Pantalla per contactar amb el club

15. Menú d'administradors

És un botó que només es troba al menú lateral i que només està disponibles pels usuaris que han iniciat sessió i són administradors. Si un usuari regular intenta entrar-hi, li sortirà un missatge del fet que no està autoritzat.

A dins del menú d'administradors podem observar una llista de botons amb diferents noms que permetran als administradors poder tenir un control absolut de totes les funcionalitats de l'aplicació des d'aquest menú.



Il·lustració 132: Pantalla del menú d'administradors

Interaccions disponibles al menú d'administradors:

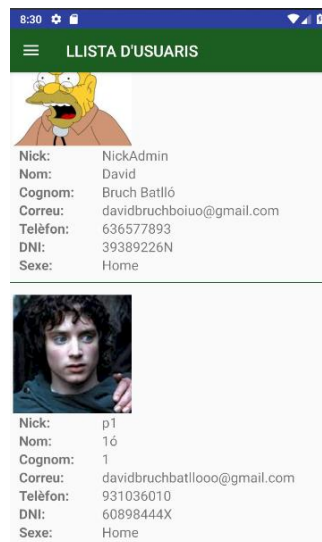
- Botó de la llista d'usuaris (1): mostra una llista de tots els usuaris del club i la seva informació incloent la imatge de perfil.
- Botó de la llista d'administradors (2): mostra una llista de tots els administradors del club i la seva informació incloent la imatge de perfil.
- Botó per eliminar un usuari (3): obre una pantalla per poder expulsar un usuari del club.
- Botó de totes les reserves (4): permet veure totes les reserves actives del club en aquell moment.
- Botó per ocupar hores (5): obre una pantalla que permet ocupar hores d'un dia i una pista concreta.
- Botó per ocupar dies (6): obre una pantalla que permet ocupar dies sencers.

- Botó de la llista negra (7): obre una llista amb la informació clau dels usuaris que han sigut expulsats prèviament.
- Botó per treure un usuari de la llista negra (8): permet treure un usuari de la llista negra perquè es pugui tornar a registrar.
- Botó del missatge de difusió (9): permet enviar un missatge de difusió per a tots els usuaris o per a tots els administradors.
- Botó per omplir o buidar torneig (10): permet als administradors omplir o buidar torneigs quan encara no estan actius perquè s'estan omplint.
- Botó per modificar partits (11): permet modificar qualsevol partit d'un torneig actiu i els seus resultats.

A continuació es mostren i s'expliquen amb més detall les diferents funcionalitats, que es poden dur a terme en el menú d'administradors, separades pels diferents botons de la llista.

15.1. Llista d'usuaris

En entrar dins d'aquesta funcionalitat es pot trobar una llista amb tota la informació de cada usuari (incloent-hi la imatge d'usuari), la qual es podrà moure amunt i avall amb el contacte i moviment del dit per si la llista és molt gran ja que varia depenen de la quantitat d'usuaris registrats.

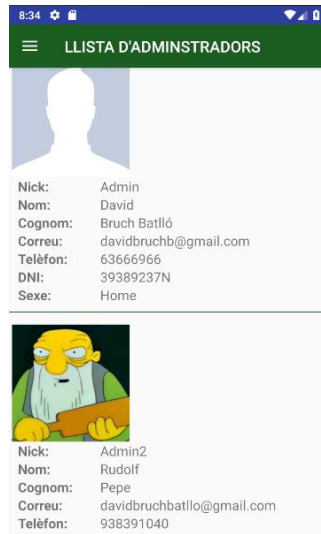


Il·lustració 133: Llista d'usuaris

Si es donés la remota possibilitat que no hi ha cap usuari sortirà un missatge especificant-ho i retorna l'administrador al menú anterior.

15.2. Llista d'administradors

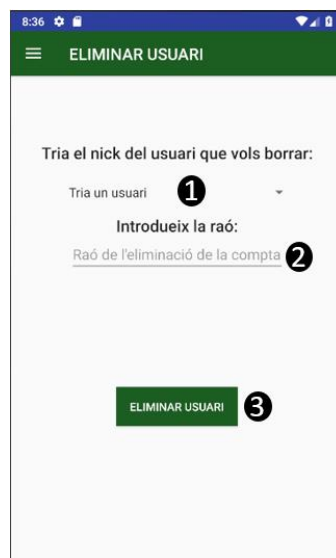
Una llista igual que l'anterior però només amb els usuaris que són administradors per poder diferenciar-los més ràpidament.



Il·lustració 134: Llista d'administradors

15.3. Eliminar usuari

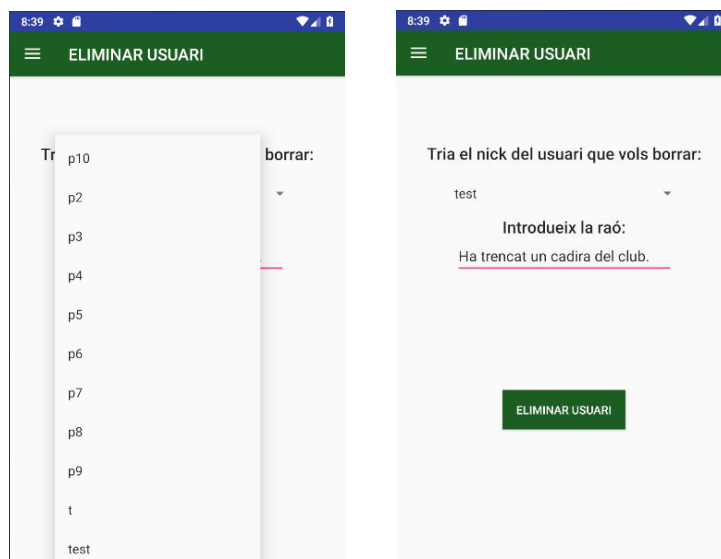
En entrar es carregen tots els noms dels usuaris i se'n pot seleccionar un de tots ells utilitzant un *spinner*. A més a més s'haurà d'introduir la raó de l'expulsió de l'usuari per guardar-la també i finalment clicar el boto per eliminar-lo i que entri dins de la "llista negra" que ja s'ha comentat amb anterioritat.



Il·lustració 135: Pantalla inicial per expulsar un usuari del club

Interaccions disponibles en eliminar un usuari:

- *Spinner* per triar un usuari (1): mostra una llista desplegable de tots els usuaris del club i se'n pot triar un.
- Quadre de text de la raó (2): quadre de text on s'ha d'introduir la raó de l'expulsió.
- Botó per eliminar un usuari (3): en clicar-se s'elimina l'usuari seleccionat i es guarda la seva informació clau a la llista negra.



Il·lustració 136: Pantalla d'expulsió amb l'spinner obert i la raó omplerta

Finalment si es tria un dels jugadors disponibles i s'entra una raó d'expulsió correctament en clicar el botó d'eliminar usuari es borra el jugador seleccionat del club i es guarden les seves dades claus a la llista negra per si es vol tornar a registrar.

15.4. Veure totes les reserves actives

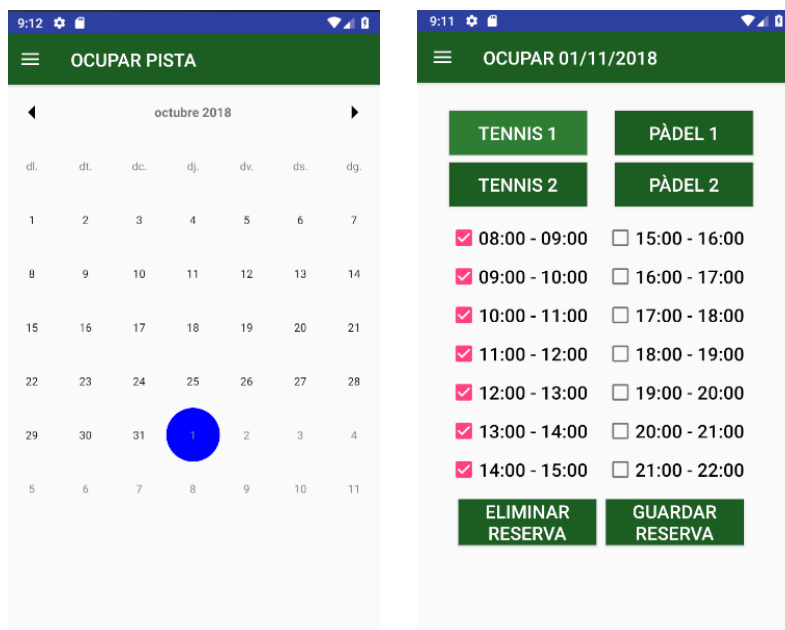
Aquesta funcionalitat serveix per veure una llista de totes les reserves actives separant les parts de la llista per dies i dins dels dies per pistes, usuaris i hores. D'aquesta manera amb una mirada rapida un administrador pot veure totes les reserves actives en aquell moment de tots els usuaris de l'aplicació.



Il·lustració 137: Llista de totes les reserves actives

15.5. Ocupar hores


Aquesta opció permet a un administrador ocupar les hores que vulgui sense limitacions en els dies i pistes que vulgui per si és festiu o s'hi ha de fer un torneig. Un cop hi entra es troba un calendari igual que per reservar una pista on et deixa seleccionar un dia. Un cop seleccionat un dia igual o posterior a l'actual (ja que si és anterior surt un missatge especificant-ho igual que per fer una reserva) s'obre una pantalla idèntica a la de reservar hores però sense les limitacions que té pels usuaris regulars.



Il·lustració 138: Calendari i pantalla per ocupar hores sense limitacions

A més a més com que un administrador té prioritat sobre els altres usuaris només veu hores ocupades si han sigut ocupades per un administrador, si no les veu lliures encara que estiguin ocupades per un usuari normal. Si l'administrador decideix ocupar hores que ja estaven ocupades per un usuari regular les pot ocupar sense problemes però es comprova internament de quins usuaris eren les hores ocupades i se'ls hi envia un avís del fet que se'ls hi han cancel·lat aquelles reserves.

Avís de cancel·lació de reserves

 Club Tennis Navàs <clubtennisn@gmail.com>
11:30

Para: davidbruchb@gmail.com

Hola p1,
Reserva cancel·lada el dia 2018-11-01 a la pista de tennis 1 a les hores:
08:00-09:00
09:00-10:00
10:00-11:00

Il·lustració 139: Correu avisant que la reserva ha sigut cancel·lada

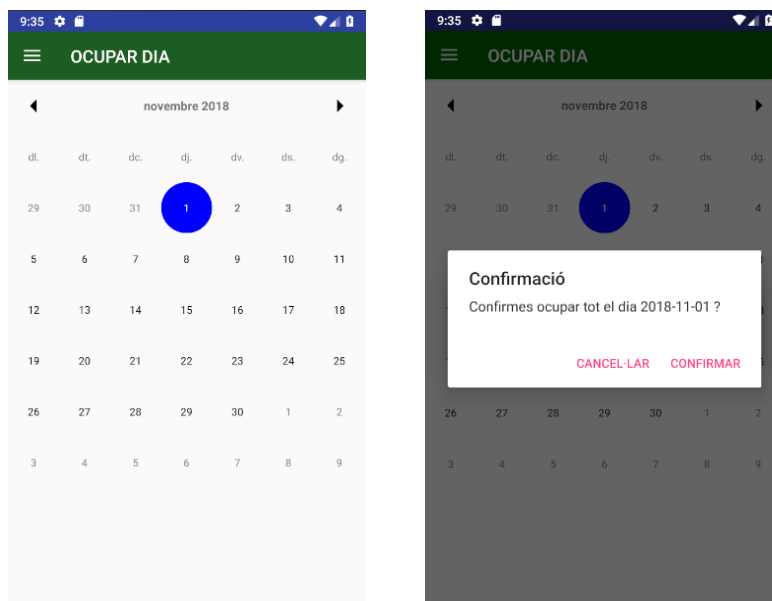
Si la reserva de l'administrador es du a terme, aquest clica el botó de reservar després d'haver seleccionat alguna hora. Igual que amb una reserva un administrador també pot eliminar les hores que havia reservat per un dia i una pista concreta amb el botó d'eliminar reserva. Els missatges d'error si s'intenta reservar sense cap hora o eliminar hores quan no n'hi ha cap són iguals que en una reserva regular. Les interaccions que es poden dur a terme són iguals a la reserva però sense les limitacions d'hores, ja que es tracta del control d'un administrador.

15.6. Ocupar dies

Dins d'aquesta opció es troba el calendari utilitzat anteriorment amb tots els dies disponibles menys els anteriors al dia actual. Si es clica un dia anterior al dia actual, apareix un missatge especificant que la data introduïda no és vàlida.

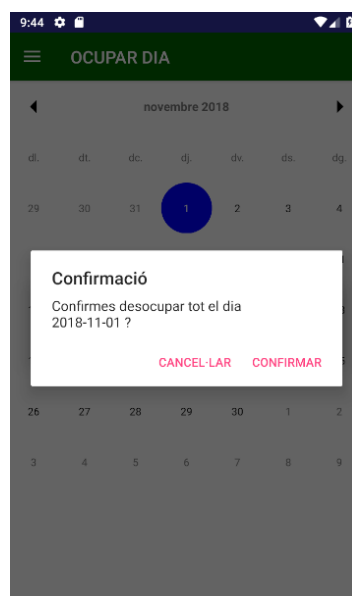
Dins d'aquesta funcionalitat clicar un dia té una dualitat depenent de si ja estava ocupat o no per un administrador. Si no estava ocupat, apareix un quadre de text preguntant si es vol ocupar tot el dia (ocupar totes les hores de totes les pistes d'aquell dia concret) amb l'opció de confirmar i

l'opció de cancel·lar. Si s'accepta, apareix un missatge confirmant que s'ha ocupat el dia correctament i si es cancel·la, no passa res.



Il·lustració 140: Quadre de text per confirmar ocupar un dia sencer

Si es clica un dia que ja estava completament ocupat per un administrador torna a aparèixer un quadre de text però preguntant si es vol desocupar el dia seleccionat. Si es confirma, apareix un missatge dient que s'ha desocupat completament aquell dia i si es cancel·la, no fa res.



Il·lustració 141: Quadre de text per confirmar la desocupació del dia sencer

Finalment remarcar que igual que quan un administrador ocupa hores a l'ocupar dies també té prioritat sobre les reserves regulars i les pot ocupar si vol. Si es dóna el cas els usuaris afectats rebran un correu electrònic especificant les hores que se li han cancel·lat de les reserves que tenia fetes.

Avis de cancel·lació de reserves



Club Tennis Navàs <clubtennisn@gmail.com>

11:41

Para: davidbruchb@gmail.com

Hola p1,

Reserva cancel·lada el dia 2018-11-01 a la pista de tennis 2 a les hores:

15:00-16:00

16:00-17:00

Reserva cancel·lada el dia 2018-11-01 a la pista de tennis 2 a les hores:

17:00-18:00

18:00-19:00

Reserva cancel·lada el dia 2018-11-01 a la pista de tennis 2 a les hores:

08:00-09:00

09:00-10:00

Reserva cancel·lada el dia 2018-11-01 a la pista de tennis 2 a les hores:

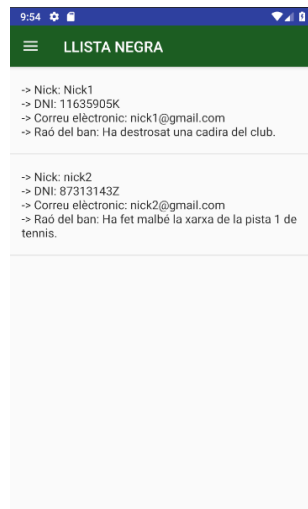
10:00-11:00

11:00-12:00

Il·lustració 142: Exemple del correu d'avis de cancel·lació de reserves

15.7. Llista negra

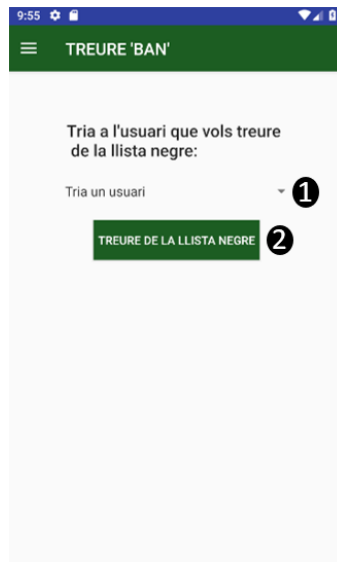
És per consultar la llista negra dels usuaris que han sigut expulsats per un administrador per alguna mala conducta en les pistes. A l'entrar en aquesta opció en trobem una llista amb la informació més rellevant de l'usuari expulsat com el "nickname", el DNI, el correu electrònic i la raó de l'expulsió. D'aquesta manera els administradors poden veure ràpidament quins usuaris s'han expulsat i per quina raó. A més a més, sempre que un usuari es registra de nou es comprova que el DNI no coincideixi amb cap dels expulsats i si coincideix, es notifica i no es deixa registrar a l'usuari.



Il·lustració 143: Pantalla de la llista negra

15.8. Treure un usuari de la llista negra

Si un usuari creu que la seva expulsió no ha sigut justa i ho pot demostrar als administradors, aquests poden entrar en aquesta pantalla per poder treure'l de la llista negra. Un cop a dins es carregaran els usuaris que estan a la llista negra i se'n podrà seleccionar un utilitzant un *spinner*. Just a sota l'usuari seleccionat hi ha un botó per dur a terme l'acció de treure'l de la llista negra.



Il·lustració 144: Pantalla que permet treure un usuari de la llista negra

Interaccions disponibles en treure un usuari de la llista negra:

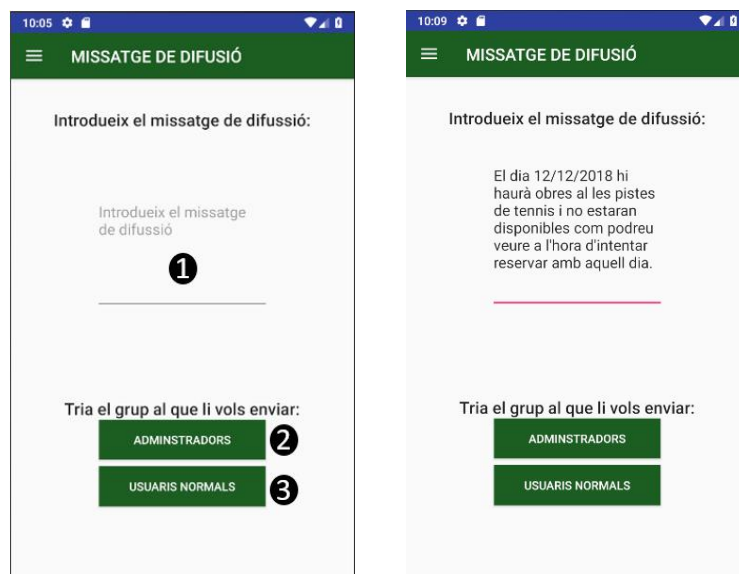
- *Spinner* per triar un usuari de la llista negra (1): mostra una llista desplegable de tots els usuaris expulsats i se'n pot triar un.
- Botó per treure l'usuari de la llista negra (2): en clicar-se s'elimina l'usuari seleccionat de la llista negra i ja pot tornar a registrar-se sense problemes.

Si no es tria cap usuari, surt un missatge d'error especificant-ho. En canvi, si es tria un usuari i es clica el botó apareix un quadre de text per confirmar o cancel·lar l'operació. En cancel·lar no passa res i en confirmar apareix un missatge dient que s'ha eliminat de la llista negra.

Finalment si s'intenta entrar dins d'aquesta opció i no hi ha cap usuari dins la llista negra surt un missatge que ho especifica i no s'entra deixant a l'administrador davant del menú amb les altres opcions disponibles.

15.9. Missatge de difusió

En aquesta pantalla hi ha un espai per introduir el missatge i a sota dues opcions diferents per poder enviar el missatge o a tots els usuaris normals (si és una notícia pública) o a tots els administradors (si és un missatge intern dels administradors). Si s'intenta clicar qualsevol dels dos botons sense haver escrit un missatge de difusió, apareix un avís que especifica que primer s'ha d'introduir un missatge.

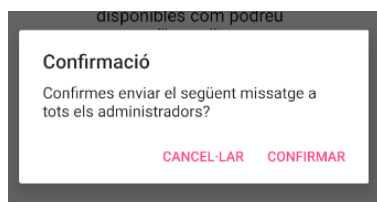


Il·lustració 145: Pantalla per enviar missatges de difusió

Interaccions disponibles pels missatges de difusió:

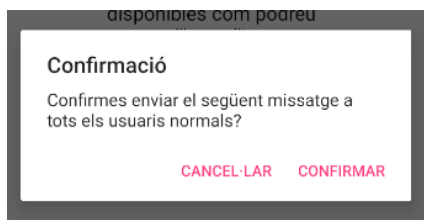
- Quadre de text per introduir el missatge de difusió (1): quadre de text on l'administrador ha d'introduir el missatge de difusió.
- Botó per enviar el missatge als administradors (2): en clicar-se s'obre un diàleg per confirmar l'enviament a tots els administradors.
- Botó per enviar el missatge als usuaris normals (3): en clicar-se s'obre un diàleg per confirmar l'enviament a tots els usuaris.

Un cop introduït bé el missatge de difusió, en clicar el botó d'administradors apareix un diàleg per confirmar o no l'acció d'enviar el missatge de difusió. Si es cancel·la, no passa res i si es confirma, s'envia el mateix missatge als diferents correus electrònics dels administradors.



Il·lustració 146: Diàleg per confirmat l'enviament als administradors

Per altra banda, si es clicca el botó d'usuaris normals, torna a aparèixer el diàleg per confirmar l'acció i si es confirma, s'envia el missatge a tots els correus electrònics dels usuaris regulars.

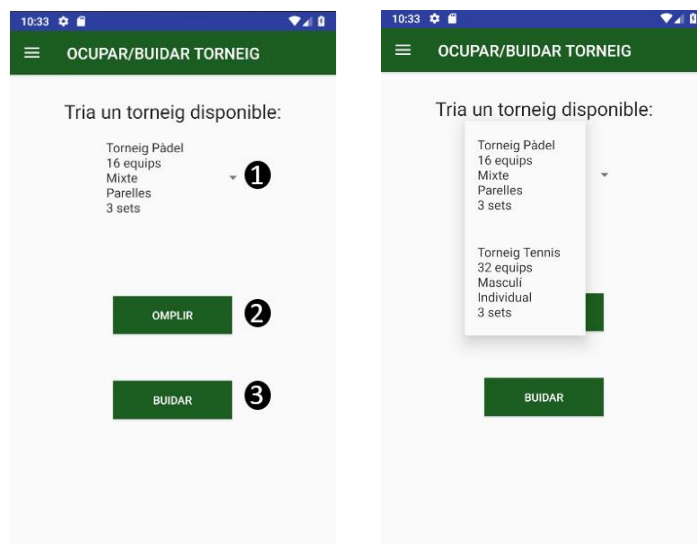


Il·lustració 147: Diàleg per confirmar l'enviament als usuaris normals

15.10. Omplir / buidar torneigs

Primer de tot s'ha de tenir en compte la diferència entre un torneig creat i un torneig actiu. Un torneig creat i prou és un torneig que ha sigut creat per un administrador i que s'està omplint de gent. Per altra banda, un torneig actiu és un torneig que ha sigut prèviament creat i que un cop s'ha omplert completament un administrador ha activat per començar el torneig.

Tenint en compte això dins d'aquesta opció només poden aparèixer els torneigs creats però no actius perquè són els que es poden omplir i buidar abans de començar els partits. Per aquesta raó, si no hi ha cap torneig creat o tots els torneigs estan actius no deixa entrar en aquesta pantalla, perquè tampoc es podria triar cap opció i apareix un missatge informant sobre això i deixant a l'administrador al menú.



Il·lustració 148: Pantalla per ocupar o buidar un torneig

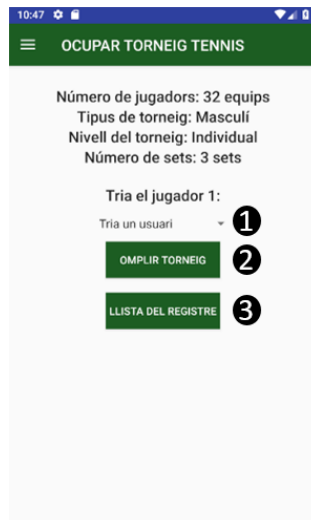
Interaccions disponibles per ocupar o buidar torneigs:

- *Spinner* per triar el torneig (1): l'administrador podrà triar entre els torneigs disponibles per poder omplir-los o buidar-los.
- Botó per omplir torneig (2): en clicar-se s'obre una pantalla nova per omplir el torneig seleccionat anteriorment.
- Botó per buidar el torneig (3): en clicar-se s'obre una pantalla nova per buidar el torneig seleccionat anteriorment.

Si sí que hi ha algun torneig creat però no actiu a l'entrar en aquesta opció es pot observar un *spinner* on es pot triar el torneig que es vol buidar o omplir d'equips. Un cop triat un torneig creat i no actiu tindrem l'opció d'omplir el torneig seleccionat o de buidar-lo.

15.10.1. Omplir torneigs

Un cop seleccionat un torneig creat i no actiu si s'ha clicat el botó d'omplir, s'obre una nova pantalla on es carrega la informació del torneig i una llista amb els usuaris per poder triar-ne un guardat en un *spinner*. Just a sota s'hi troba el botó per dur a terme l'opció d'omplir el torneig i un altre botó per veure la llista dels participants d'aquest torneig.



Il·lustració 149: Pantalla per omplir el torneig

Interaccions disponibles per ocupar un torneig:

- *Spinner* per triar el primer jugador (1): si el torneig és individual, l'usuari actual serà l'únic participant de l'equip però si és per parelles, apareix un diàleg de text per introduir el segon integrant igual que amb el registre normal als torneigs.
- Botó per omplir torneig (2): en clicar-se s'omple el torneig seleccionat anteriorment per l'usuari seleccionat al *spinner*.
- Botó de llista del registre (3): en clicar-se s'obre una pantalla on es pot veure la llista dels equips registrats amb la seva informació.

Si es clica el botó d'omplir però no s'ha seleccionat cap usuari o se n'ha seleccionat algun que ja participava en aquest torneig apareix un missatge personalitzat recordant-ho a l'administrador.

El botó d'omplir reacciona diferent depenent de si el torneig és individual o per parelles. Si és individual, apareix un diàleg per introduir el nom de l'equip on es pot confirmar o cancel·lar el nom. Si el nom d'equip ja existeix o es deixa en blanc apareix un missatge d'avís personalitzat. En canvi, si és per parelles, abans del diàleg per triar el nom d'equip apareix un diàleg per introduir el nom del segon integrant de l'equip. Si el nom del segon integrant no existeix dins del club, es deixa en blanc, es posa el mateix nom que el primer jugador de l'equip o ja participa en un altre equip apareix un missatge personalitzat, depenen del problema.

Un cop s'han introduït bé les dades que demana, tant per individual com per parelles es completa el registre de l'equip dins del torneig seleccionat prèviament. Finalment, per millorar la comoditat dels administradors es pot utilitzar el botó de la llista de registrats, que canvia depenent del torneig seleccionat anteriorment. Aquesta llista és la mateixa que es pot observar si entres a través del torneig que s'està omplint.

15.10.2. Buidar torneigs

Un cop seleccionat un torneig creat i no actiu, si s'ha clicat el botó de buidar, s'obre una nova pantalla on es carrega la informació del torneig i una llista amb els equips registrats que apareixen en l'*spinner*. Just a sota s'hi troba el botó per dur a terme l'opció de buidar el torneig i un altre botó per veure la llista dels participants d'aquest torneig.



II·lustració 150: Pantalla per buidar el torneig seleccionat anteriorment

Interaccions disponibles per buidar un torneig:

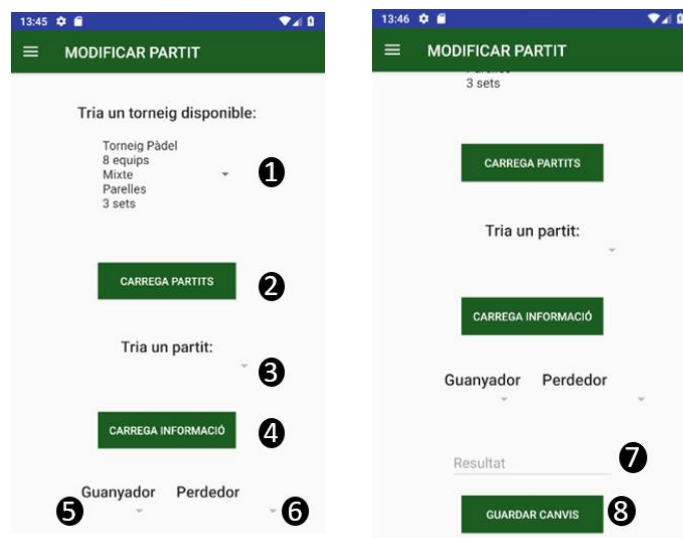
- *Spinner* per triar un equip (1): l'usuari pot triar un dels equips ja registrats al torneig per expulsar-lo.
- Botó per buidar torneig (2): en clicar-se es buida el torneig seleccionat anteriorment, treient l'equip seleccionat.
- Botó de llista del registre (3): en clicar-se s'obre una pantalla on es pot veure la llista dels equips registrats amb la seva informació.

Si es clica el botó de buidar torneig però no s'ha seleccionat cap equip un missatge personalitzat recordant-ho a l'administrador. En canvi, si se selecciona un equip dels disponibles i es clica el botó de buidar, s'elimina l'equip seleccionat i apareix un missatge confirmant que s'ha eliminat correctament.

Finalment, igual que en la pantalla per omplir, també hi ha l'opció d'anar directament a la llista d'equips registrats del torneig seleccionat prèviament amb l'últim botó de la pantalla.

15.11. Modificar partits

Per poder entrar dins d'aquesta opció hi ha d'haver com a mínim un torneig creat i actiu completament omplert i activat per un administrador. Si no hi ha cap torneig creat i actiu apareix un missatge d'avís informant sobre això, ja que si no hi ha cap torneig actiu, també vol dir que no hi ha cap partit creat.



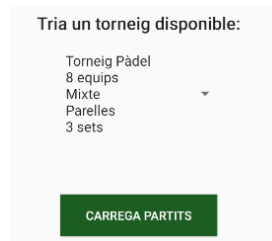
Interaccions disponibles per buidar un torneig:

- *Spinner* per triar un torneig actiu (1): l'administrador pot triar un dels torneigs actius per modificar qualsevol dels seus partits.
- Botó per carregar els partits (2): en clicar-se es carregen els partits del torneig seleccionat al *spinner* de sota i s'habilita perquè es pugui utilitzar.
- *Spinner* per triar un partit (3): permet triar un partit del torneig seleccionat anteriorment per modificar-lo.
- Botó per carregar la informació del partit (4): en clicar-se traspasa la informació pel partit als 2 *spinners* (guanyador i perdedor) i al quadre de text del resultat. A més a més, habilita l'última part de la pantalla.
- *Spinner* del guanyador (5): és on es mostra el guanyador actual del partit seleccionat i permet modificar-lo.
- *Spinner* del perdedor (6): és on es mostra el perdedor actual del partit seleccionat i permet modificar-lo.
- Quadre de text del resultat (7): s'hi mostra el resultat actual del partit seleccionat i permet modificar-lo.
- Botó per guardar els canvis (8): permet guardar els canvis que s'hagin pogut fer en el guanyador, perdedor o resultat.

Per altra banda, si hi ha algun torneig actiu, entrarem sense cap problema a la nova pantalla per modificar els partits. Aquesta pantalla està separada en 3 zones diferenciades pels 3 botons que hi ha. En entrar només la primera zona estarà disponible, ja que les altres dos no estan habilitades i no es poden clicar en entrar per primer cop.

15.11.1. Carregar partits del torneig seleccionat

En aquesta part, que és l'única amb la qual es pot interactuar quan entres al principi, hi ha un "spinner" on es carregen els torneigs actius en aquell moment. L'administrador ha de triar un torneig dels disponibles i clicar el botó de carregar partits per passar a la següent zona de la pantalla.



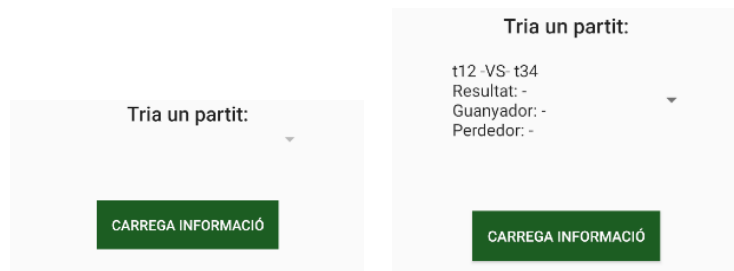
Il·lustració 152: Primera zona per modificar partits

Si l'usuari no tria cap torneig, se selecciona automàticament el primer de la llista disponible i com que per entrar sempre hi ha d'haver com a mínim un torneig actiu sempre hi haurà com a mínim una opció disponible.

15.11.2. Càrrega de la informació dels partits del torneig carregat anteriorment

Un cop carregats els partits disponibles es poden seleccionar en el segon "spinner", ja que la segona zona a passat a estar habilitada per poder-hi interactuar. En aquesta part, es poden veure dins del "spinner" tots els partits del torneig seleccionat a la zona 1 amb la informació més rellevant d'aquests (els dos equips que juguen, el guanyador, el perdedor i el resultat). El guanyador, el perdedor i el resultat apareixen amb un "-" si no s'han registrat encara.

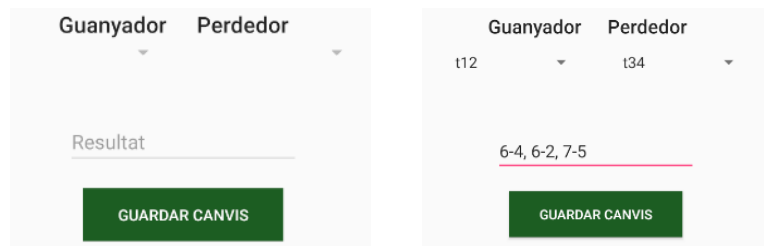
Com que només poden aparèixer torneigs actius sempre hi hauran partits disponibles, ja que com a mínim ja s'haurà generat la primera ronda de partits.



Il·lustració 153: Segona zona no habilitada i habilitada

15.11.3. Càrrega de la informació del partit seleccionat

Un cop clicat el botó anterior per carregar la informació s'habilita l'última part de la pantalla on es carrega el guanyador i el perdedor en *spinners* separats i el resultat en un quadre de text editable. D'aquesta manera l'administrador pot veure com està el partit actualment i pot canviar qualsevol de les 3 opcions (guanyador, perdedor o resultat).



Il·lustració 154: Tercera zona no habilitada i habilitada

L'administrador pot fer els canvis que vulgui sense cap limitació en els partits, però sí que poden aparèixer missatges d'avís quan s'intenta fer alguna operació no lògica. Per exemple, si s'intenta que un equip guanyi i perdi al mateix temps apareix un missatge per avisar del fet que no es pot fer. A més a més, si es clica el botó de guardar canvis però no s'ha fet cap canvi també apareix un avís sobre això i no s'envia cap petició.

Finalment si s'ha fet algun canvi en el partit i es clica el botó apareix un missatge confirmant el canvi i torna a no habilitar les zones 2 i 3 deixant només l'opció de tornar a clicar un torneig actiu per carregar els nous partits.

Annex B: Funcionalitats d'Android

Dins del programa *Android* s'utilitzen diferents funcionalitats interessants d'explicar per les possibilitats que brinden a l'aplicació. S'explica amb detall el funcionament de les activitats i els fragments, com s'implementen en la interfície gràfica i els seus cicles de vida. A més a més, també s'expliquen els diferents elements que s'han utilitzat a la interfície gràfica i que permeten interactuar amb l'usuari.

En aquesta aplicació s'han utilitzat tant activitats com fragments per descriure el comportament de les pantalles depenent de les interaccions i estímuls de l'usuari.

1) Activitats

Una activitat permet crear una pantalla amb un disseny específic per la interfície d'usuari i normalment aquesta pantalla ocupa tota l'àrea disponible o se sobreposa a una altra. Moltes aplicacions diferents estan formades per múltiples activitats que s'interconnecten entre ells de manera flexible. Normalment sempre hi ha una activitat principal que s'acostuma a mostrar a l'usuari com un panell de notícies o un menú principal per a poder accedir a les altres funcionalitats.

Qualsevol activitat pot iniciar-ne una altra per poder canviar de pantalla o fer alguna funcionalitat específica. Quan hi ha el canvi d'una activitat cap a una altra, la primera no es destrueix, sinó que s'emmagatzema a una pila que permetrà tornar-hi si l'usuari clica el botó per tirar enrere. A més a més, quan es produeix un canvi en una activitat es notifica el canvi mitjançant mètodes de tipus *callback*, que permeten crear, aturar, reprendre o destruir les activitats per separat. Gràcies a això, es poden realitzar diferents tasques específiques segons el canvi d'estat de l'activitat.

Els dos mètodes, dins d'una activitat, més importants que cal destacar són l'*onCreate()* i l'*onPause()*. El mètode *onCreate()* sempre s'ha d'implementar, ja que és el que crea l'activitat des de 0 i inicialitza els components gràfics de la pantalla corresponent. Dins d'aquest mètode és on es crida la funcionalitat *setContentView(id XML)* que rep com a paràmetre l'identificador del *layout* (pantalla gràfica) en format XML, que és l'arxiu que defineix el disseny de la interfície gràfica.

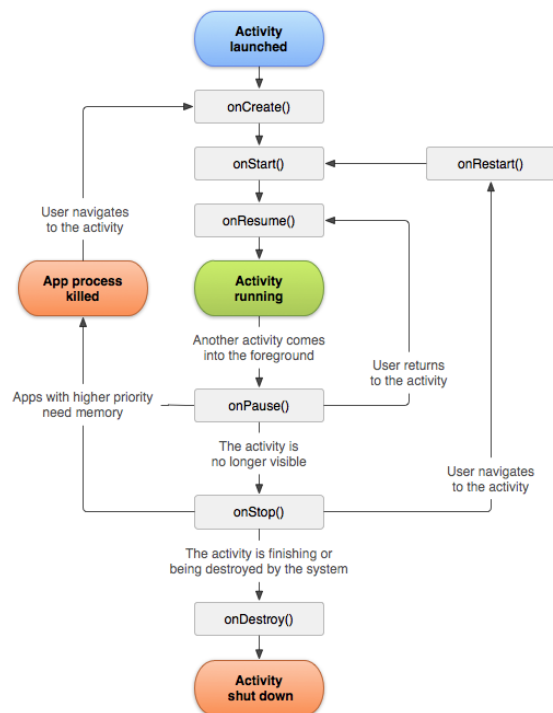
Per altra banda, el mètode *onPause()* és molt important perquè s'executa cada cop que s'abandona l'activitat actual. Quan es produeix un canvi d'activitats s'ha de verificar quines variables i funcionalitats s'han de conservar per si l'usuari torna a l'activitat inicial. Les activitats tenen un cicle de vida dividit principalment en 3 estats diferenciats amb diferents funcionalitats cada un.

Una activitat té essencialment quatre estats:

- Si una activitat es troba en primer pla de la pantalla (a la part superior de la pila), està activa o s'executa.
- Si una activitat ha perdut l'enfocament però encara és visible (és a dir, una nova activitat no completa o transparent té un focus a la part superior de la vostra activitat), es troba en pausa. Una activitat pausada és completament viva (manté tota la informació de l'estat i membre i roman adherida al gestor de finestres), però pot ser assassinada pel sistema en situacions extremes de memòria baixa.
- Si una activitat està completament fosca per una altra activitat, es deté. Encara conserva tota la informació de l'estat i dels membres, però, ja no és visible per l'usuari, per la qual cosa la seva finestra està oculta i sovint serà assassinada pel sistema quan es necessita memòria en altres llocs.
- Si una activitat està en pausa o es deté, el sistema pot deixar anar l'activitat de la memòria demanant-li que finalitzi, o que simplement mori el procés. Quan es torni a mostrar a l'usuari, s'ha de reiniciar i restablir completament el seu estat anterior.

El següent diagrama mostra les vies d'estat importants d'una activitat. Els rectangles quadrats representen mètodes de devolució de trucada que podeu implementar per realitzar operacions

quan l'activitat es mou entre estats. Els rectangles arrodonits de colors són els principals estats en els quals pot participar l'activitat.



Il·lustració 155: Diagrama d'estats importants d'una activitat. Font: [Externa](#)

Hi ha tres cicles principals que poden dur a terme les activitats:

- El temps complet d'una activitat que passa entre la primera crida a *onCreate(Bundle)* fins a una sola trucada final a *onDestroy()*. Una activitat farà tota la configuració de l'estat "global" a *onCreate()* i alliberarà tots els recursos restants a *onDestroy()*.
- La vida útil visible d'una activitat que passa entre una crida a *onStart()* fins a una crida corresponent a *onStop()*. Durant aquest temps, l'usuari pot veure l'activitat a la pantalla, tot i que pot ser que no estigui en primer pla i interactui amb l'usuari.
- La vida útil de primer pla d'una activitat, que passa entre una crida a *onResume()* fins a una trucada corresponent a *onPause()*. Durant aquest temps, l'activitat es troba davant de totes les altres activitats i interactuant amb l'usuari.

Finalment, s'ha de tenir en compte que les activitats han d'estar declarades a l'arxiu *Manifest*, ja que sinó no seran accessibles pel sistema de cap manera. En aquest arxiu també s'haurà d'especificar quina és l'activitat principal i quina activitat s'executa primer al iniciar l'aplicació.

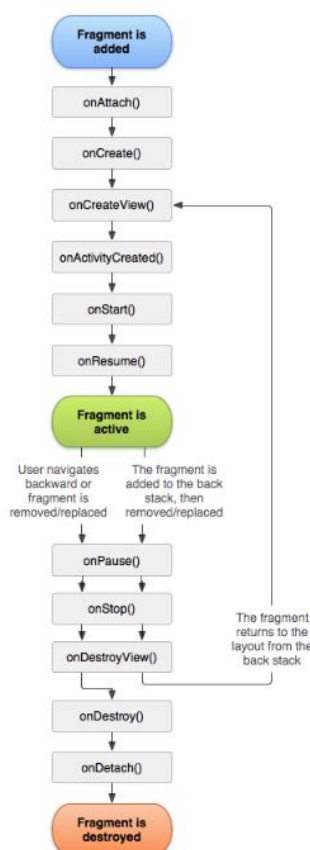
2) Fragments

Un Fragment representa un comportament o una part de la interfície d'usuari en una activitat. Pots combinar múltiples fragments en una sola activitat per crear un menú i tornar a fer servir un fragment en múltiples activitats.

Es pot pensar en un fragment com una secció modular d'una activitat que té el seu cicle de vida propi i que rep els seus propis esdeveniments que pugui tenir d'entrada. Un fragment sempre ha d'estar integrat a una activitat i el cicle de vida del fragment es veu directament afectat pel cicle de vida de l'activitat amfitriona. Per exemple, quan l'activitat està

pausada, també ho estan tots els seus fragments, i quan l'activitat es destrueix, el mateix passa amb tots els fragments.

No obstant això, mentre una activitat s'està executant (està en l'estat del cicle de vida represa), pots manipular cada fragment de forma independent. Quan fas una transacció de fragments com aquesta, també pots afegir a una pila d'activitats administrada per l'activitat; cada entrada de la pila d'activitats en l'activitat és un registre de la transacció de fragments realitzada. La pila d'activitats li permet a l'usuari invertir una transacció de fragments (navegar cap enrere) en pressionar el botó enrere.



Il·lustració 156: Cicle de vida d'un fragment. Font: [Externa](#)

Podem destacar principalment el mètode `onCreateView()`, que el sistema ho crida quan el fragment ha de dissenyar la seva interfície d'usuari per primera vegada. Per dissenyar una IU per al teu fragment, has de tornar una `View` des d'aquest mètode que serà l'arrel del disseny del teu fragment. En cas que no hi hagi disseny gràfic pots tornar nul i no proporciona una IU.

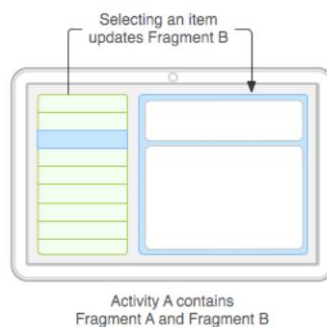
Per a poder administrar correctament els fragments s'utilitza el `FragmentManager`, que ens permet realitzar diferents funcionalitats:

- `addBackStack()`: afageix un fragment a la pila en estat pausat i d'aquesta manera permet retornar al fragment anterior clicant el botó d'enrere.
- `popBackStack()`: treu un fragment de la pila per tal d'executar-lo.
- `findFragmentById()`: obté fragments que existeixen en l'activitat actual i que disposen d'interfície gràfica a través d'una `id`.
- `findFragmentByTag()`: obté fragments que poden tenir o no interfície gràfica a través d'un sobrenom.

En tot moment, el *FragmentManager* també permet la utilització de transaccions entre fragments o entre l'activitat i els fragments. Es pot generar una instància de la classe *FragmentManager.Transaction*, que mitjançant les funcions *remove()*, *add()* i *replace()* es poden gestionar diferents transaccions entre fragments. La comanda *commit()*, és la que du a terme els canvis i s'ha de cridar sempre com a l'últim mètode de la transacció.

3) Menú lateral desplegable

En aquesta aplicació s'ha dissenyat un menú desplegable que es pot obrir i és visible des de qualsevol pantalla de l'aplicació per poder fer que la navegació al llarg de l'aplicació sigui molt més fàcil i fluida per l'usuari. Per aquesta raó s'han hagut d'utilitzar fragments per poder fer que la pantalla principal sigui una pantalla en blanc juntament amb el menú i a partir dels fragments poder canviar la vista de la pantalla depenent del fragment que es cridi.



Il·lustració 157: Esquema d'una activitat que conté 2 fragments. Font: [Externa](#)

Tal com es pot observar en la imatge, la idea del menú desplegable és la mateixa. El menú i una pantalla en blanc són l'activitat principal i totes les altres pantalles són fragments que s'intercanvien per canviar la vista de la pantalla.

A continuació, es mostren els principals components que integren els dissenys gràfics utilitzats:

Buttons: els botons són una icona normalment rectangular de mida i color variable amb un text a dins que l'usuari pot clicar per interactuar amb ell i iniciar un esdeveniment.

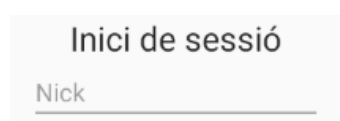
ImageButton: es una icona de mida variable on s'hi pot inserir una imatge que serà completament clicable, igual que un botó normal.



Il·lustració 158: Exemple d'un botó i un boto imatge

EditText: requadre que permet introduir text o modificar-lo. Al crear l'objecte s'ha d'especificar el format que pot acceptar, com per exemple, correus, contrasenyes, text pla, etc. Normalment es mostra el text per defecte, per tal d'indicar al usuari que s'hi ha de posar, amb l'atribut *android:hint*.

TextView: text pla que mostra l'aplicació no modificable directament per l'usuari.



Il·lustració 159: Exemple de un text pla i un quadre de text

RadioButton: permet seleccionar diverses opcions al mateix temps. Per exemple, a l'hora de seleccionar diverses hores al mateix temps per fer una reserva.

Spinner: proporciona un menú desplegable que permet triar una de les opcions. Si no s'ha triat cap opció es mostra un valor definit.

The screenshot shows a mobile app interface. On the left, there is a grid of time slots for reservation, each with a checkbox. The selected times are 08:00 - 09:00, 09:00 - 10:00, 18:00 - 19:00, and 19:00 - 20:00. On the right, there is a spinner menu titled 'Tria un usuari' (Select a user) with a 'borra' (delete) button. The spinner shows a list of users: el 7872628, NickAdmin 39389226N, opo 11635905k, p1 60898444X, p10 4535, p2 222222, p3 333333333, p4 4444444, p5 5555555, and n6.

II·lustració 160: Exemple de RadioButton i d'Spinner

Dialeg: es generen en temps d'execució i mostren un quadre de text personalitzable per confirmar alguna acció o introduir informació.

The image shows two examples of dialog boxes. The first dialog box is titled 'Confirmació' and asks 'Confirmes eliminar totes les reserves del usuari actual?' (Do you confirm deleting all reservations of the current user?). It has two buttons: 'CANCEL·LAR' and 'CONFIRMAR'. The second dialog box is also titled 'Confirmació' and asks 'Confirmes el registre al torneig de tennis amb el nom d'equip actual?' (Do you confirm the registration for the tennis tournament with the current team name?). It has a text input field containing 'team1' and two buttons: 'CANCEL·LAR' and 'CONFIRMAR'.

II·lustració 161: Exemple de 2 diàlegs diferents

ListView: permet emmagatzemar informació en format de cadena de caràcters en les diferents cel·les d'una llista gràfica. A més a més, si la llista és molt llarga, es pot navegar per ella sense problemes fent arrossegant el dit amunt i avall per la superfície de la pantalla.

RecyclerView: permet emmagatzemar qualsevol tipus d'informació i de components gràfics a cada cel·la d'una llista gràfica. En el cas d'aquest projecte s'ha utilitzat per afegir la imatge de cada jugador juntament amb la seva informació. Es pot navegar a través de la llista de la mateixa manera que amb el *ListView*.